Преглед НЦД 10 (2007), 23–32

Julijana Mirčevski, Biljana Đokić
(Informatička škola "Educon", Beograd),
Nikola Popović
(Ministarstvo inostranih poslova, Beograd)

THE WAVELET TRANSFORM BASED SOFTWARE SUITABLE FOR A DIGITAL CONTENTS ANALYSIS

Abstract: The digital contents database requires a more complexes software tools in order to searching, analysis's, compression and reproduction. Wavelet theory provides the fast discrete algorithms suitable to computer programming application. The clear mathematical theory is a good base to creating a programs environment and a number programs packages. The wavelet transform application are thinking less without a programming software package but the most used are WaveLab, LastWave, MegaWave i Rice Wavelet Toolbox. Because the mentioned software products belong to free ware software category, it is possible to use that with registration legality. The answer on the questions is given in the paper: which wavelets, which models, which programs, which date sets are giving the best results.
Keywords: digitalized contents, searching, wavelet transformation, free ware software

1. Introduction

Wavelet theory provides the fast discrete algorithms suitable for applications in computer programming. A mathematical analysis alone is often unable to predict fully the behavior and suitability of algorithms for specific signals. Experiments are necessary and such experiments ought to in principle be reproducible, just like experiments in other fields of sciences. The reproducibility of experiments requires the complete software and full source code for inspection, modification and application under varied parameter settings. The testing results of the existing programs library as WaveLab, LastWave, MEGAWAVE and Rice Wavelet Toolbox are presented in this paper. The testing procedure was executed in the area of laser beam scattering image and in information retrieval procedure.

Jon Claerbout established the concepts of really reproducible research in the "Computational Sciences". The "Computational Sciences" he has in mind [1], are fields in which mathematical, computer science heuristics may suggest algorithms to be tried on scientific signal processing, and imaging problems, but mathematical analysis alone is not able to predict fully the behavior and suitability of algorithms for specific datasets. Therefore, experiments are necessary and such experiments ought, in principle, to be reproducible, just as experiments in other fields of science.

Suppose we are working in an area like exploration seismology where the goal is an image of the subsurface, and computational science has aim to produce the best images. However, the deliverable is not an image itself, but instead the software environment that, applied in the right way, produces the image, and which could be applied to other datasets to produce equally nice images. The scientific findings are depended of knowledge *of parameter settings* for this complex software environment that seem to lead to good results on real datasets.

With this as background, reproducibility of experiments requires deposing with the complete software environment available in other laboratories and the full source code available to control, modification, and application under varied parameter settings.

This concept is denoted as a very significant for the wavelet community. According to Claerbout, perception that as we approach specific applications using wavelets, we are becoming a computational science like seismic imaging or like laser scattering imaging. Arguments over data compression have everything to do with specifics: exactly what was done (which wavelets, which coders, which corpus of data was compressed) with exactly what parameters. In this setting, publishing figures or results without the complete software environment could be compared to publishing an announcement of a mathematical theorem without giving the proof [2].

With the recent rapid spread of Internet facilities worldwide and the standardization of scientific computing on about five machine architectures, most of which are UNIX-based, it becomes feasible and timely to actually experiment with protocols implementing the goal of reproducible research.

The software based on the wavelet transform is a modest step in this direction. Authors of these programs are making available code, which will allow the interested researchers to reproduce the figures in published articles and to study the exact parameter settings and algorithms, which were used in those articles.

2. Freeware concept

Richard Stallman [9] and others associated with the GNU project have pioneered the idea of free software - software that can freely be redistributed by users. This does not mean free of cost; it means the rights any one person has over the software are the same as those of any other.

Authors of WaveLab software are influenced by this, but obviously cannot completely follow them since they require by users that they have MATLAB, which is not freely redistributable.

Modern scientific computing depends on *quantitative programming environments* like MATLAB, S-PLUS, MATH-EMATICA, X-MATH, IDL and so on. These are widely available, widely understood high-level languages in which key concepts of scientific discourse (Fast Fourier Transform, etc.) are available as built-in, easily usable features. Now, according to our opinion, a complete freeware software implementation of one's computational experiments requires, writing all one's tools from scratch in **C** or **Java.** Scientists in some application area are using the massive size of the datasets but most working scientists have smaller-scale datasets. They cannot be expected to start from scratch and develop all their code in C when they can get, much more quickly and reliably, the same results in a very high-level language.

An applied of Wavelet transformation cannot be thinking without software tools and the most used are: **WaveLab, LastWave, MegaWave** i **Rice Wavelet Toolbox**. The newer investigation and advanced user application promoted the software packages as **Virtual Reality Modeling Language – VRML** and **Smart Information Retrieval Systems.**

Above-mentioned program packages belong to the freeware software category so it is possible to use these packages under legality registration circumstances. There is an especial situation by programs, which are implemented in Matlab that is not freeware. However, Matlab usual already exist in such research laboratories.

3. Searching, analysis, compression and reproduction

Limited storage space and transmission through narrow band-with channels create a need for compressing signals [3], while minimizing their degradation. Transform codes compress signals by decomposing them in an orthonormal basis. Major applications are data storage and transmission through channels with a limited bandwidth [4].

A transform coder decomposes a signal in an orthogonal basis and quantizes the decomposition coefficients. The distortion of the restored signal is minimized by optimizing the quantization, the basis and the bit allocation.

Searching a particular term in a document requires forming his term signal. A term signal is a sequence of values that indicate the occurrence of a searched term in particular section of a document. Here, should be presented an example the term signals obtaining [5]. Let us with t signed a term in document d, then term signal can be represented by

$$\tilde{f}_{d,t} = [f_{d,t,0}, f_{d,t,1}, \ldots f_{d,t,N-1}]$$

where $f_{d,t,n}$ is the value of the signal component. Let we have N signal components and K terms in the document. The value of the n-th component can be calculated as the occurrences of term t between the (nK/N)–th word and the {(n+1)K/N-1}-th word in the document. If the term signal has, e.g. eight components, $f_{d,t,0}$ would contain the number of times term t occurred in the first eight of document d . For whole document should be N=1, and $f_{d,t,0}$ would contain the count of term t throughout the whole document.

4. WaveLab software tools

WaveLab is a library of MATLAB routines for wavelet analysis, wavelet-packet analysis, cosine-packet analysis and matching pursuit. The library is available free of charge over the Internet. Versions are provided for Macintosh, UNIX and Windows machines.

WaveLab has over 1100 files, which are documented, indexed and cross-referenced in various ways. MATLAB MEX files are used extensively to increase throughput. In addition to routines implementing basic wavelet transforms for finite data sets (both periodic transforms and boundary-corrected transforms), wavelet-packet analysis, cosine-packet analysis and matching pursuit, the library contains scripts which the authors believe will assist in learning the practical aspects of wavelet analysis:
Scripts that reproduce the figures in the authors' published articles, including the de-noising articles of Donoho and Johnstone [1].
``Workouts" that give a quick guide to wavelets (1-d and 2-d); wavelet analysis; wavelet synthesis; wavelet and cosine packets, including the Coifman-Wickerhauser best-basis methodology; matching pursuit; and applications such as data expansion, progressive data transmission, image compression, speech segmentation, de-noising, fast matrix multiplication in wavelet bases, etc.

WaveLab also offers:

- A library of datasets that is easily accessible to the user. Besides artificial signals that have scientific or pedagogical appeal, real data ranging from an image of Ingrid Daubechies to a recording by Enrico Caruso are included.
- A point-and-click browser that allows the user to select data, perform various transforms or de-noising operations, and then see the results without using the MATLAB command-line interface.
- Extensive documentation, including on-line documentation for each function, Contents files for each subdirectory, a *Reference Manual*, an *Architecture Guide* and an overview document, *About WaveLab that* introduces the software to a first-time user.

WaveLab package was tested under Windows XP in Mat Lab 5.3 and Mat Lab 6.0 environment. It was shown simpler connection and better functionality with MatLab5.3. Some functions in Wave Lab program system, particular in graphic mode are already exist in Mat Lab environment but the Wave Lab modules shows the better functionality and simpler application in image analysis system.

5. LastWave program environment

LastWave is a program tools that makes a wavelet signal and image processing environment implemented in **C** language for **UNIX, Windows and Macintosh** platforms. This package belongs to the stand-alone freeware software category, that means, not require any additional commercial software. Last Wave can be retrieved at the internet address: http://wave.cmap.polytechnique.fr/soft/LastWave

LastWave core is composed of six packages. The ***kernel, signa***l and ***image*** packages are ***C***-packages and the ***disp, misc*** and ***terminal*** packages are defined in script.

The ***kernel*** package corresponds to the core of ***LastWave***. It mainly includes

- the parser and the structure of the command language
- basic commands
- definition of the basic types such as strings (&str), numbers (&num,&float,&int), list of values (&listv, i.e., list of object of arbitrary type) and some other.
- basic graphic management

The ***signal*** package defines the signal structure (of type &signal) and includes all the commands that deal with signals. It also includes the definition of the graphic class GraphSignal that allows displaying signals.

The ***image*** package includes the image structure (of type &image) and includes all the commands that deal with images and/or matrices (thus it includes the linear algebra commands). It also includes the definition of the graphic class GraphImage that allows displaying images

The ***disp*** package defines the command disp that allows to display basically anything!

The ***terminal*** package defines the behavior of the terminal (help system, completion,...). To run the help system it can be type ***Help*** at startup.

The ***misc*** package defines miscellaneous things including how to get interactive help on graphic objects

LastWave_2_0_4 package was installed and tested under UNIX /System V, SUSE Linux 9.3 and SUSE Linux 10.0. Unix/System V as a server system is shown some problems but SUSE Linux 9.3 system was very „user friendly“ during installation LastWave_2_0_4 package and then in running particular function modules.

6. Megawave philosophy

The aim of MegaWave2 [7] is to make the coding of signal and image-oriented algorithms easier. An algorithm is implemented as a function (or a set of functions) written in C language; such a function (or set of functions) is called a module.

The programmer does not write a complete program: what a module becomes is the matter of the ***MegaWave2*** compiler. This compiler adds input/output code to generate a run-time command: the module's command. The module can then be called under the shell as ***UNIX*** commands.

The source of a module is a file (its name is the module name followed by the ***.c*** extension) which contains pure ***C*** instructions. It begins with a header put into comments (/* ... */), so the ***C*** compiler ignores it). However, the MegaWave2 preprocessor decodes this header. Information is defined about the module, so are the author's name, the version number, etc. More important are the information about the usage, that is, about the input/output objects (or input/output variables) of the module. This usage may say that a given variable (for example a ***float***) is an optional input with default value 1.0, and that another variable is an output.

After the header comes the regular C body, where functions are defined. One function must always be present, it is the main function of the module and it must have the same name as the module. When we refer to input/output objects of the module, we think about the parameters of this main function. Only the main function is global (i.e. can be accessed from other modules), all other functions are local.

In the module's skeleton below, the main function is module and the input/output objects are obj1,obj2,...:

```
/*------------------ MegaWave2 Module ----------------*/
/* mwcommand
.
. [header, including usage of obj1,obj2,...]
.
*/
internal_function(a1,a2,....)
.
. [definition of this function]
.
module(obj1,obj2,....)
{
  .
  .
  internal_function(b1,b2,...);
```

```
  .
  .
}
```

A module can access to the main function of any other module, according to the following rules:

Public modules - Such modules are available for all users; they are located in the MegaWave2 system directory (for example $MEGAWAVE2).

Private modules - All MegaWave2 users coding new modules generate their own private modules (located in the directory $MY_MEGAWAVE2). A module cannot access to another private module from another user. Therefore, it is necessary to know, if somebody has written a module of some interest for others, the MegaWave2 administrator should make it public.

Each module belongs to a group. A group puts together all modules dealing about the same subject. For example, user can imagine a group fourier where all the algorithms about the Fourier transform are put. However, also user may want to distinguish between Fourier applied on one-dimensional signals and Fourier applied on images. User can define subgroups of the group fourier, as for example signal and image: should be get two groups named fourier/signal and fourier/image.

The meaning of the input/output objects is in following explanation. First, they are variables of C type. All scalar types are allowed, such as char, int, long, float and other. Types that are more sophisticated are available. They are called MegaWave2 memory (or internal) types, and they are always pointers to a structure. The structure defines the object to be processed by the module. For example, the memory type Cimage represents monochrome images with gray levels of (unsigned) Char values. The memory type Curve represents a discrete curve in the plane, etc. In ``MegaWave2 System Library'' there are a description of all available MegaWave2 memory types.

These objects live outside a module on the following way. They are C variables until the process finishes: a module which calls another module just gives the objects as parameters of the function; into the XMegaWave2 interpreter, all commands reside in memory and therefore objects remain C variables until they are removed. When the process finishes (e.g. at the end of a module's command), the output objects have to be saved on disk as a file. The same problem occurs when a process begins: input objects have to be read from files. The format of the file depends on the memory type, we call it the MegaWave2 file (or external) type. Whereas there is only one memory type associated to an object, a memory type may be represented on disk with many different file types. This is because MegaWave2 recognizes several standard file formats, especially for the images. Also, see Volume two for a description of all available MegaWave2 file types.

MegaWave2 is a free Unix/Linux package that includes the following items:

1. a library of image processing modules that contains, in addition to classical tools, original algorithms written by researchers to conduct the experiments reported in theirs articles. In this way, it contributes to the reproducibility of research.

2. a ***C*** preprocessor and a system library that allows easy and fast development of new modules : only the algorithms have to be implemented, then the inputs/outputs

are automatically handled by the preprocessor, and documentation (module syntax) is generated as well.

Below is an example of a simple module, that rotates a float image using a bilinear interpolation. The main part of the program is devoted to the algorithm: only a short header describes the inputs/outputs of the module and only one line handles memory allocations.

```
/*--------------------------- Commande MegaWave ----------------------------*/
/* mwcommand
   name = {frot};
   version = {"1.0"};
   author = {"Lionel Moisan"};
   function = {"Rotate a Fimage using bilinear interpolation"};
   usage = {
 'a':[a=0.0]->a  "rotation angle (in degrees, counterclockwise, default 0.0)",
 'b':[b=0.0]->b  "background grey value (default: 0.0)",
 'k'->k_flag     "to keep original input image size (keep center position)",
 in->in          "input Fimage",
 out<-out        "output Fimage"
   };
*/
#include <stdio.h>
#include <math.h>
#include "mw.h"
void bound(x,y,ca,sa,xmin,xmax,ymin,ymax)
int x,y;
float ca,sa;
int *xmin,*xmax,*ymin,*ymax;
{
   int rx,ry;

   rx = (int)floor(ca*(float)x+sa*(float)y);
   ry = (int)floor(-sa*(float)x+ca*(float)y);
   if (rx<*xmin) *xmin=rx; if (rx>*xmax) *xmax=rx;
   if (ry<*ymin) *ymin=ry; if (ry>*ymax) *ymax=ry;
}
   /********** Compute new image location **********/
   if (k_flag) {
     /* crop image and fix center */
     xmin = ymin = 0;
     xmax = nx-1;
     ymax = ny-1;
     xtrans = 0.5*( (float)(nx-1)*(1.0-ca)+(float)(ny-1)*sa );
     ytrans = 0.5*( (float)(ny-1)*(1.0-ca)-(float)(nx-1)*sa );
   } else {
   /********** Rotate image **********/
   for (x=xmin;x<=xmax;x++)
     for (y=ymin;y<=ymax;y++) {
```

```
    xp = ca*(float)x-sa*(float)y + xtrans;
    yp = sa*(float)x+ca*(float)y + ytrans;
    x1 = (int)floor(xp);
    y1 = (int)floor(yp);
    ux = xp-(float)x1;
    uy = yp-(float)y1;
    adr = y1*nx+x1;
    tx1 = (x1>=0 && x1<nx);
    tx2 = (x1+1>=0 && x1+1<nx);
    ty1 = (y1>=0 && y1<ny);
    ty2 = (y1+1>=0 && y1+1<ny);
    a11 = (tx1 && ty1? in->gray[adr]:*b);
    a12 = (tx1 && ty2? in->gray[adr+nx]:*b);
    a21 = (tx2 && ty1? in->gray[adr+1]:*b);
    a22 = (tx2 && ty2? in->gray[adr+nx+1]:*b);
    out->gray[(y-ymin)*sx+x-xmin] =
      (1.0-uy)*((1.0-ux)*a11+ux*a21)+uy*((1.0-ux)*a12+ux*a22);
  }
```

This and some other modules we modified and make suitable to our application in information retrieval procedure.

7. Rice Wavelet Toolbox

The Rice Wavelet Toolbox (RWT) is a collection of **Matlab** M-files and C MEX-files for 1D and 2D wavelet and filter bank design, analysis, and processing. The toolbox provides tools for denoising and interfaces directly with our ***Matlab*** code for wavelet domain hidden Markov models and wavelet regularized deconvolution.

It is important to know [8] that earlier version of the ***Rice Wavelet Toolbox*** (up to version 2.3) do not compile under ***Matlab*** versions 6.0 (Release 12) and above. The current version (version 2.4) fixes this compatibility issue. It is necessary to point at some differences between existed versions. Version 1.1 from 1993, was the first officially released **RWT.** Version 2.01 from 1994 had several significant enhancements.

The current distribution, Version 2.4 (Dec 1, 2002) that is tested and used, has been streamlined and packaged for different systems, including Solaris, Linux, and Microsoft Windows. We are tested under SUSE Linux 9.3. Some functions omitted in Version 2.4 can be found in the Version 2.01 distribution. The ***Rice Wavelet Toolbox*** is open source software under license. It is distributed as free of charge and without warranty.

Source code in gzip compressed format for **UNIX** users. To unpack gzip'ed files, execute "***gunzip rwt.tar.gz***" followed by ***"tar -xvf rwt.tar"***.

Source code in PC zip compressed format for PC users the source code contains matlab ***.m*** files and ***.c*** files that need to be compiled.

We must pay attention on the differences in pre-compiled packages under various operating systems (i.e. Linux version 7.2 compiled using Matlab 6.5 (R13) and Linux Red Hat Release 9.0, Windows NT/2000/XP compiled using Matlab 6.5 (R13) and Windows XP, etc.). Authors called the RWT program users to report bugs or send comments regarding the toolbox to address webmaster-dsp@ece.rice.edu.

8. Conclusion

The wavelet transform is a tool which has been used in many areas of science and engineering to extract information about the self-similarity of signals. Using this information, it makes possible to encode the signals in more compact manner and easily extract desired content to perform other tasks as a compare signals, find frequency and position of term appearances and other.

From the computer specialist people point of view, more complex tools, suitable interfaces, short execution time, are the performance to be favorable by researchers.

From the user's point of view, migration of systems to new environment, tools and procedures always provoke an opposition and feeling of estrangement the source materials , which there are under user carrying.

The efficient computation, registration legality and possibility to compare the investigation results with other people worldwide, [9] are giving the motivation to rich the using rules of the software packages based on the wavelet transform, such as ***WaveLab***, ***LastWave*** and other.

The solution can be found in easy accessible procedure to setting parameters computation. Just in wavelet domain is appeared the real multidisciplinary approach in process design, structuring and implementation software packages like as WaveLab, LastWave and others.

Our experience with mentioned programs shows the impossibility of a general and simple comparison. WaveLab package is the oldest software product based on the wavelet transform and today, is the most complex program tools in wavelet field. There is necessity for MatLab software in background and it hass the less good features of WaveLab package. LastWave does not require any additional commercial package, an extensive on-line documentation is available and showed good results in image compression with a wavelet transform code and reconstruction of the coded image. MEGAWAVE is very efficacious in local cosine processing [9] with sound and image applications. RWT indicates as a very suitable to denoising treatment with orthogonal and biorthogonal transforms but require MatLab, also.

The essential conclusion could be: an extensive usage of these program tools and their modules to compound a new appropriate program tool for retrieving date bases of digitalized contents is needed and important.

References

1. http://www.wavelet.org
2. Stephane Mallat, *A Wavelet Tour of Signal processing*, Academic Press, San Diego, 1999.
3. Stephane G. Mallat, *A theory for multiresolution signal decomposition: the wavelet representation*, IEEE Transactions on pattern analysis and machine intelligence 11(7), July 1989, 674–693
4. About WaveLab, http://www-stat.stanford.edu/~wavelab

5. Laurence A.F. Park, Kotagiri Ramamohanarao and Marimuthu Palaniswami, *A novel document retrieval method using the discrete wavelet transform*, ACM Transaction on information systems 23(3), July 2005, 267–298.
6. http://wave.cmap.polytechnique.fr/soft/LastWave
7. http://www.cmla.ens-cachan.fr/Cmla/Megawave/
8. http://www-dsp.rice.edu/software/RWT
9. J. Mirčevski, M. Srećković, S. Bojanić, *Some views on the software support to the quantum electronics research*, Proc. Lasers 1995, STS Press, McLEAN, 275–281, Charleston, 1995
10. Julijana Mirčevski, Željka Tomić, Milesa Srećković, Ljubomir Vulićević, *Wavelet transformacija – pogodnost za programiranje*, Naucno-strucni simpozijum INFOTEH-JAHORINA 2006, Srpsko Sarajevo, mart 2006
11. http://www.wikipedia.org/wiki/RichardStallman

СОФТВЕР ЗАСНОВАН НА ВЕЈВЕЛЕТ ТРАНСФОРМАЦИЈИ ПОГОДАН ЗА АНАЛИЗУ ДИГИТАЛИЗОВАНОГ САДРЖАЈА

Сажетак. Базе података дигитализованог садржаја захтевају сложеније програмске алате за претраживање, анализу, компресију и репродукцију. Вејвелет теорија обезбеђује брзе дискретне алгоритме погодне за апликације у рачунарском програмирању. Јасна математичка теорија добра је основа за креацију програмских окружења и бројних програмских пакета. Примена вејвелет трансформације незамислива је без програмских алата а најчешће коришћени су WaveLab, LastWave, MegaWave и Rice Wavelet Toolbox. Будући да наведени програмски производи спадају у категорију бесплатног софтвера могуће је њихово коришћење уз легалитет регистрације. У раду се дају одговори на питања: који вејвелети, који модели, који програми, који скупови података дају најуверљивије резултате.
Кључне речи: дигитализовани садржаји, претраживање, вејвелет трансформација, бесплатни софтвер

julijana@afrodita.rcub.bg.ac.yu
djokicb@eunet.yu