# PARALLELIZATION OF AN IMAGE COMPRESSION AND DECOMPRESSION ALGORITHM BASED ON 1D WAVELET TRANSFORMATION

*S. Khanfir, M. Jemni & E. Ben Braiek*

Ecole Supérieure des Sciences et Techniques de Tunis
5, Av. Taha Hussein, B.P. 56, Bab Mnara 1008, Tunis, TUNISIE
sami.khanfir@esstt.rnu.tn, mohamed.jemni@fst.rnu.tn, ezzedine.benbraiek@esstt.rnu.tn

## ABSTRACT

Wavelet analysis has received considerable interest in the recent years because of its efficiency in the several practical applications. Image processing for wavelet transformation is considered as one of the most powerful methods that provides a good quality of results. However, its implementation may be too time-consuming accordingly to the problem size. Parallel processing can be a solution to speed up wavelet transformation programs. In this context, and in order to have a quick image compression/decompression program based on 1D wavelet transformation, we have designed three parallel algorithms that where implemented on an IBM RS6000/SP machine. The first parallel algorithm exploits control parallelism it was developed with OpenMP and executed on one four-processor node. The two others exploit data parallelism and were developed with MPI directives. Finally, we present an evaluation of these algorithms based on an experimental study.

## 1. INTRODUCTION

During the last years, and with the appearance of the wavelet, there was a growing interest for the use of this efficient processing method in various applied areas.

It has been proved that it is a powerful tool for signal and image processing i.e. Compression/Decompression, [2, [3, [4, [5]. Even if it is a good solution for many problems in this area, it is a source of other problems. The most important one is the computing time of these algorithms: the larger the image is, the higher the processing and compressing treatment is, from time computing point of view. That's why the use of parallel computing can be an adequate solution to minimize execution time.

In this paper we present three parallel algorithms in addition to the serial one for image Compression/Decompression. These algorithms were programmed for an IBM RS6000/SP multiprocessor composed of 8 four-processor nodes, where the processors of each node share a same memory.

The first parallel algorithm exploits the control parallelism inherent in the serial one. It is based on the parallelization of iterations that don't carry any loop dependencies.

This algorithm, developed with OpenMP, is dedicated to a shared memory parallel machine, thus we executed it on one node of our machine.

The second one is a data parallel algorithm; it is based on the decomposition of the image matrix into a number of equal sizes sub-matrices. Each sub-matrix can be treated independently by one processor and at the end a communication step, based on MPI directives, is achieved. However, this latter version cannot be considered as a parallelized version of the serial algorithm because, it is not semantically equivalent to it due to the borders effects produced by the data distribution. Remark that this effect is not visible at weak compression rates, but by comparing the sequential and parallel results value by value, we may find differences. The third parallel algorithm is also a data parallel one, but we used a different way of data distribution such that the parallel execution gives exactly the same result as the serial one.

## 2. IMAGE COMPRESSION & DECOMPRESSION BASED ON 1D WAVELET TRANSFORM ALGORITHM

Wavelet offers an alternative approach, for signal and image processing; its most important quality is that it is highly localized in time and scale. Its main application is in image compression with very good rate-distortion performances [3].

### 2.1. Image Decomposition and Reconstruction Algorithms Based on 1D Wavelet Transforms

In 2-D, the wavelet representation can be computed with a pyramidal algorithm which can be seen as a one-dimensional wavelet transform along the x and y axes [6]. It can be proved that a two-dimensional wavelet transform can be computed with a separable extension of the one-dimensional decomposition algorithm [7], these algorithms (decomposition and reconstruction) are illustrated by the two block diagrams depicted in Fig. 1 and Fig. 2.

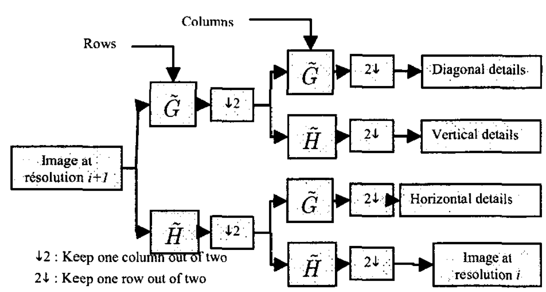



Fig. 1. Decomposition of an image in wavelet coefficients with the 1D wavelet transform algoritm

0-7803-8379-6/04/$20.00 ©2004 IEEE.

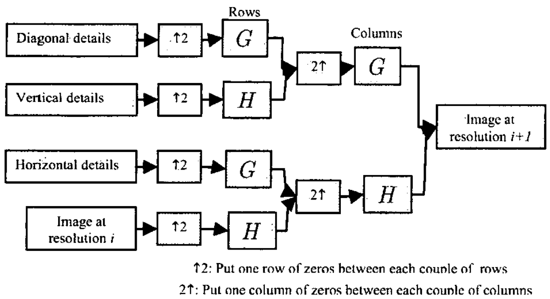



Fig. 2. Reconstruction of an image from its wavelet coefficients with the 1D wavelet transform algoritm

### 2.2. Image Compression using 1D Wavelet Transform

The compression of the image is practiced after the decomposition of the image in wavelet coefficients by the 1D DWT algorithm and it consists of three steps:

i. Thresholding: putting coefficients that are lower to a fixed threshold $\varepsilon$ (by the necessities of problem treated), which is responsible of the rate of the compression and the fidelity of the image to the original one (the higher $\varepsilon$ is, the higher compression is, and the less the image is similar to the original one).
ii. Quantification: before saving the compressed image, we must convert the floating-point coefficients into integer numbers.
iii. Coding: the last step consists of coding the resulting numbers.

## 3. SERIAL ALGORITHM FOR IMAGE COMPRESSION AND DECOMPRESSION BASED ON 1D DISCRETE WAVELET TRANSFORM

### 3.1. Decomposition algorithm

The decomposition algorithm of a signal in wavelet coefficients (Algo/1) is described as follows:

i. The filter $\tilde{H}$ will multiply its coefficients, term per term, with those of the signal.
ii. Arriving at the end of the filter, there will be a summation of the different obtained products and the result will be stored in a vector.
iii. The filter $\tilde{G}$ will do the same things as the filter H except for the storage which will be done in the second half of the vector.
iv. The two filters will translater over the signal, using a step 2, and repeat steps (i, ii and iii).
v. Repeat steps (i, ii, iii and iv) until the end of the signal.
vi. The resulting vector (resulting from the decomposition above) will replace the original signal.
vii. Repeat the last 6 steps on the signal until reaching the desired level of the decomposition.

To generalize this algorithm to the image decomposition we must do the following steps:

i. Treat the different rows of the image using (Algo/1).
ii. Treat the different columns of the image using the same algorithm.

### 3.2. Image Compression Algorithm[1]

The image compression procedure (Algo/2) is illustrated by the following phases:

i. Fix the threshold $\varepsilon$ of the compression.
ii. Compare the wavelet coefficients of the image with $\varepsilon$ and set them to zero if they are lower than it.

### 3.3. Reconstruction Algorithm

The reconstruction process (Algo/3) is nearly identical to the decomposition algorithm described above but we must replace the filters $\tilde{H}$ and $\tilde{G}$ by H and G filters and begin from signals at resolution i to signals of higher resolution.
To generalize this algorithm to the image decomposition we must do the following steps:

i. Treat the different rows of the image using (Algo/3).
ii. Treat the different columns of the image using the same algorithm.

## 4. PARALLEL ALGORITHMS FOR IMAGE COMPRESSION AND DECOMPRESSION BASED ON 1D DISCRETE WAVELET TRANSFORM

### 4.1. Parallel Algorithm using Open MP

This parallel algorithm is based on dependence analysis of loop iterations of the serial algorithms Algo/1, Algo/2 and Algo/3. Every loop which does not carry dependences between its iterations can be parallelized by OpenMP directives.

#### *4.1.1. Decomposition Algorithm for Image Compression Based On 1D Wavelet Transform Using Open MP*

```
FOR j = n-1, 0
        DO ALL l = 0, 2^(j+1)-1
            w[l]=0
        END DO ALL
        DO ALL l = 0, 2^j-1
                i = (2 * l + min_hd) mod 2^(j+1)
                for k = 0, len_hd-1
                        w[l] = w[l] + hd[k] * v[i]
                        i = ( i + 1 ) mod 2^(j+1)
                end for
                i = ( 2 * l+min_gd ) mod 2^(j+1)
                ls = l + 2^j
                for k = 0, len_gd-1
                        w [ls] = w [ls] + gd[k] * v[i]
                        i = ( i + 1 ) mod 2^(j+1)
                end for
        END DO ALL
```

[1] In the following image compression algorithm, we will not consider the quantification and coding steps.

```
        DO ALL l = 0, 2^(j+1)-1
                v[l] = w[l]
        END DO ALL
END FOR
```

#### 4.1.2. *Parallel Image Compression Algorithm*

The parallel compression phase is traduced in these steps

i. Fix the threshold ε of the compression.
ii. All the possessors will divide the task of comparing the wavelet coefficients of the image with ε and put them to zero if they are lower than it.

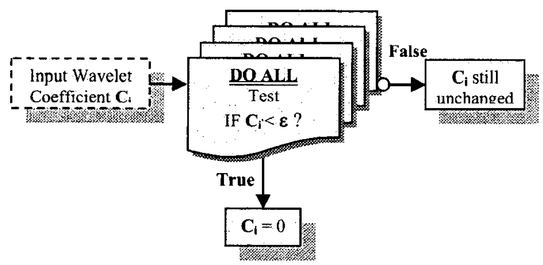



Fig. 3. Parallel Image Compression Algorithm

#### 4.1.3. *Reconstruction Algorithm for Image Decompression Based On 1D Wavelet Transform Using Open MP*

```
For j = 1, n
        DO ALL l = 0, 2^j-1
                w[l] = 0
        END DO ALL
        DO ALL k = 0, 2^j-1
                i = (floor (( k - min_hp ) / 2 )) mod 2^(j-1)
                if i < 0
                i = 2^(j-1) + i
                lb = ( k - min_hp ) mod 2
                For l = lb, len_hp-1, pas 2
                        w[k] = w[k] + hp[l] * v[i]
                        i = ( i – 1 ) mod 2^(j-1)
                end for
                if i < 0
                i = 2^(j-1)
                i = (floor (( k - min_gp ) / 2 )) mod 2^(j-1)
                if i < 0
                i = 2^(j-1)
                lb = ( k - min_gp ) mod 2
                for l = lb, len_gp-1 pas2
                    w [k] = w [k] + gp [l] * v [i + 2^(j-1)]
                    i = ( i -1 ) mod 2^(j-1)
                end for
                if i < 0
                i = 2^(j-1)
        END DO ALL
        DO ALL l = 0, 2^j-1
                v[l]=w[l]
        END DO ALL
END FOR
```

### 4.2. Parallel Algorithms Using MPI

#### 4.2.1. *Parallel Algorithm Using MPI Based On The Decomposition Of The Image In identical Blocs*

This parallelization method is a little bit different from the earlier, in fact the main difference resides in the decomposition method of the image (see Fig. 4); which is divided in homogeneous square blocks, this means that the image can be decomposed into 4, 16 or 64... parts.

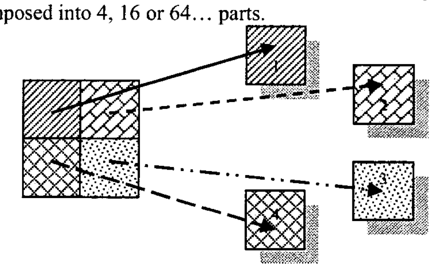


Fig. 4. Image decomposition in 4 blocs

##### 4.2.1.1. *Decomposition And Compression Algorithm With The Different Communication Phases*

i. Decomposition of the image into an identical number of bloks (see Fig. 4).
ii. Communication phase between processors: the master processor will send the needed data for all the other slave processors.
iii. Treatment phase: every processor will proceed to the treatment of its block of data by applying the decomposition algorithm (Algo/1) and then the compression one (Algo/2).
iv. Communication phase: all the processors will send their information to the master. This later reconstructs the compressed image when it receives all data.

##### 4.2.1.2. *Decompression Algorithm*

The image decompression process is identical to the decomposition and compression algorithm described above but we must replace algorithm (Algo/1) by Algorithm (Algo/3) and omit the compression phase.

#### 4.2.2. *Parallel Algorithm Using MPI Based On The Decomposition Of The Image In Line and column Blocks*

We present now a Decomposition/Compression Algorithm with the different communication phases (see Fig. 5)

i. Decomposition of the original matrix in row blocks.
ii. Transmission of the different line blocks to the processors.
iii. Every processor will treat its line block by the algorithm (Algo/1).
iv. Redistribution of the data to the different processors in column blocks.
v. Every processor will treat its column block by the algorithm (Algo/1).
vi. Compression of the column blocs by using the algorithm (Algo/2); every processor will treat its column block separately.

vii. Reconstruction of the compressed image in the memory of the master processor.

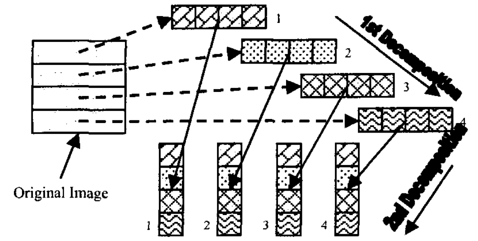
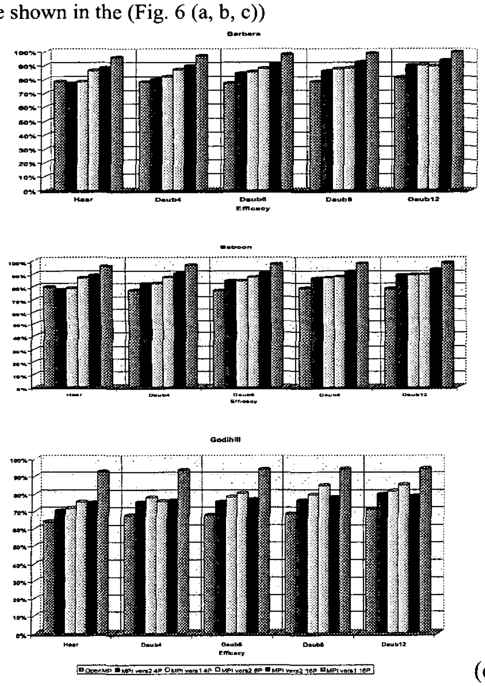



Fig. 5. Image decomposition[2] in line and column blocs

If we consider the image decompression process, this one is identical to the decomposition/compression algorithm described above but we must replace algorithm (Algo/1) by the Algorithm (Algo/3) and omit the sixth phase.

## 5. EXPERIMENTAL EVALUATIONS

In order to evaluate the efficiency of our algorithms, we have run a series of experimentations on an IBM RS6000/SP. Tests are carried out on the well known images "Barbara, Baboon and Godlhill" we have applied various filters i.e. Haar, Daubechies 4, Daubechies 6, Daubechies 8 and Daubechies 12, and results are shown in the (Fig. 6 (a, b, c))



(a)

(b)

(c)

Fig. 6 Efficiency of the different algorithms tested on the different images

[2] This illustration is totally simplified, the different phases are much more complex

## 6. CONCLUSION

In this paper, we have presented three parallel algorithms for image Compression/Decompression based on wavelet transformation. The first algorithm is implemented with OpenMP directives for a parallel shared memory machine, whereas the two others are data parallel algorithm implemented with MPI directives. The performance evaluation shows that in general, MPI versions give a little bit better speedups. This may be justified by the fact that with MPI the programmer exploits manually inherent parallelism nevertheless with OpenMP this exploitation is done by the compiler. Concerning the MPI versions, the experimental study shows that version1 gives better speedups because version 2 requires more communications. On the other hand version2 generates results that are identical to the serial programs. It is a parallelized version of the serial algorithm and thereby it is semantically equivalent to this latter, whereas version 1 represents another algorithm witch is directly designed in parallel.

## 7. REFERENCES


[1] Björn Jawerth and Wim Sweldens, "An Overview of Wavelet Based Multiresolution Analyses", AMS (MOS) classification: 42-02, 42C10

[2] Daniel Lemire and Alain Béliveau, "White Paper sur la Compression d'Images par la Fast Wavelet Transform", Compression d'Images en réseaux et Automatismes, CIRA, http://www.ondelette.com/CIRA/whitepaper CIRA_(Francais)_2_2.pdf, 2000.

[3] Julien Reichel, "Integer Wavelet Transform for Embedded Lossy to Lossless Image Compression", IEEE Transactions on Image Processing, Vol. 10, NO. 3, March 200.

[4] Michael B. Martin and Amy E. Bell, "New Image Compression Techniques Using Multiwavelets and Multiwavelet Packets", IEEE Transactions on Image Processing, Vol. 10, N0.4, April 2001.

[5] Rafael C. Gonzalez and Richard E. Woods, "Digital Image Processing", Addison-Wesley Publishing Company.

[6] Stephane G. Mallat, "Theory for Multiresolution Signal Decomposition: The Wavelet Representation", IEEE Transactions On Pattern And Machine Intelligence, Vol. 11, No.7, July 1989.

[7] Yves Meyer, "Ondelettes and Algorithmes concurents", Colloque GRETSI – JUAN-LES-PINS, Septembre 1993.