PARALLEL IMPLEMENTATION OF IMAGE CODING USING WAVELET TRANSFORM: SYNDEX SOFTWARE ENVIRONMENT APPLICATION

Christophe Cudel[*], *Bertrand Vigouroux*[**]
[*] LAM - Equipe de Troyes - LTI - IUT de TROYES - BP 396 - 10026 TROYES CEDEX - FRANCE
Tel: (33) 25.42.46.43 - e-mail: cudel@altern.com
[**] IUT d'ANGERS - BP 2018 - 49016 ANGERS CEDEX - FRANCE
Tel: (33) 41.73.53.20 - e-mail: bertrand.vigouroux@univ-angers.fr

ABSTRACT

This work is a contribution to Adequation between Algorithm and Architecture. It presents an example of application made with SynDEx, a software environment to implement signal processing or automatic algorithms on multi-processor network. This communication shows that a Conditioned Data Flow Graph used for modelling an algorithm, is enough to do an implementation on multi-processeur network.

1 INTRODUCTION

Algorithms implementation on transputers network may be quite difficult, and takes a long time to be developed when algorithms are complex, especially if real-time constraints are required. In this communication, we propose to use SynDEx, a help implementation software, made by Y.SOREL and C.LAVARENNE, working at INRIA (Institut National de Recherche en Informatique et Automatique, Rocquencourt, France), to realize an image compression algorithm using wavelet transform, on a transputers network [1]. This paper presents SynDEx and the CDFG, image compression with wavelets transform, and the implementation on a transputer network.

2 SYNDEX PRESENTATION

SynDEx is an help tool used for implementation of signal processing algorithms on parallel calculators. It performs an Adequation between Algorithm and Architecture from algorithm and architecture graphs model [2]. Algorithm graph is a Conditioned Data Flow Graph (CDFG). This model puts forward algorithm parallelism. Architecture graph is simply described by a graphic representation of processor network, showing how they are connected between each other (see figure 1).

With this 2 graphs, SynDEx generates the C code, for the multi-processors architecture (multi-transputers or multi-DSP).

2.1 Conditioned Data Flow Graph (CDFG)

This model excludes the hardware characteristics of the network processors (processors type and communication time). A CDFG is made of tops (algorithm actions), and arcs joining the tops (data flow: scalars, vectors, or matrices).

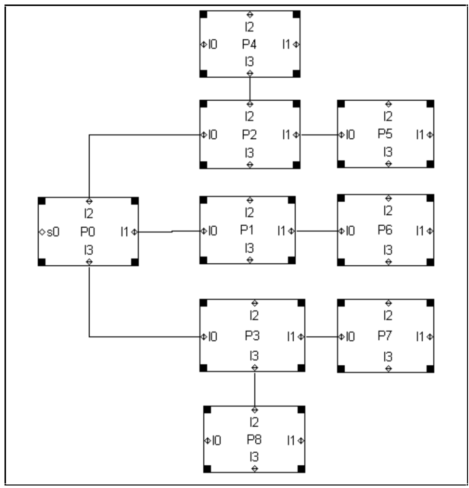

Figure 1 : transputer network

CDFG is running when data are at the graph input. So, a CDFG is a synchronous representation of an algorithm [3]. This representation needs only 4 elementary functions for describe an algorithm. To define these functions, a clock P(X) is joined up with each data flow. The clock is TRUE if the signal is present, FALSE if the signal is not. The clock T(B), corresponding to the TRUE value of a boolean signal is also defined. These definitions allow to put forward the 4 elementary functions: immediate function, delay function, sampling function, and mixing function. The characteristics of these functions are given on table 1. Immediate function and delay function have no effect on data flow clock. Therefore determinist algorithms can be described only with them. The 2 other functions will allow to explain conditioned algorithms. Each user build his graph for explain the maximum of parallelism.

2.2 SynDEx environment

SynDEx adequation between algorithm and architecture, realised with glutton algorithm, optimizes the placing of the GFDC tops on the processors. The C code produced, manages the placing of the algoritms tops and communications between the processors.
For the moment, with the 3.6 version, the code generated, is dedicated to transputer network (INMOS T800) and multi-DSP (Texas Instrument TMS320C40).

Elementary functions	Relations
Immediat function	Y=f(X) P(Y)=P(X)
Delay	Y(n)=X(n-i) P(Y)=P(X)
Sampling	Y=X $P(Y) = P(X) \cap T(B)$
Mixing	Y=X0 or Y=X1 $P(Y) = P(X0) \cap P(X1)$

Table 1: 4 elementary functions for a GFDC

3 IMAGE COMPRESSION WITH WAVELETS

A fundamental goal of data compression is to reduce the bit rate for transmission or storage while maintaining an acceptable fidelity or image quality. Compression can be achieved by transforming the data, projecting it on basis functions, and then encoding this transform.
The wavelet transform formulated in the eighties by Y.Meyer [4], knows an efficient implementation with the pyramidal algorithm given by S.Mallat [5].
This algorithm projects an image at different levels of resolution. For each iteration, an image corresponding to the resolution level m-1, is performed in 4 sub-images. The first sub-image is the low resolution level m, and the others are the details (horizontal, vertical and diagonal) lost between the resolution levels m-1 and m.
The compression consists of quantizing the details with Vector Quantization (VQ), and coding low resolution sub-images with a Differential Pulse Code Modulation (DPCM).

3.1 Vector Quantization (VQ)

According to Shannon's rate distortion theory, better results are always obtained when vectors rather than scalars are quantized. Developed by Gersho and Gray in 1980, vector quantization has proven to be a powerful tool for digital image compression [6]. The method is encoding a sequence of samples (vectors) rather than encoding each sample individually. Encoding is performed by approximating the sequence to be coded by a vector belonging to a catalogue of shape, usually known as a codebook or dictionary (see figure 4).
The codebook is created and optimized using the LBG classification algorithm with a mean squared error (MSE) criterion. This algorithm is designed to perform a classification based on a training set constitued of vectors belonging to different images; it converges iteratively toward a locally optimal codebook.

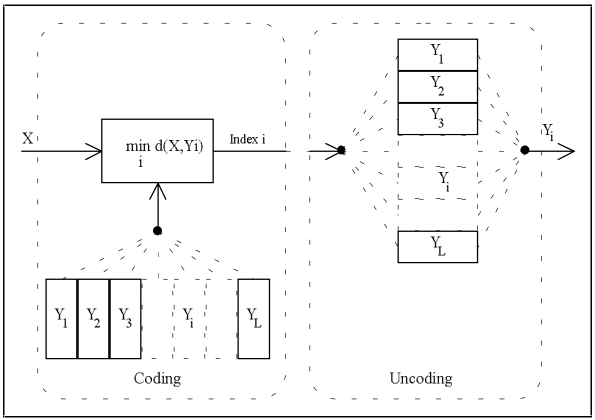

Figure 2: Vector quantization

3.2 Differential Pulse Code Modulation (DPCM)

In a general predictive coding scheme, the correlation between the neighbouring pixel value is used to form a prediction for each pixel [7]. By far, the most common approach to predictive coding is differential pulse code modulation (DPCM). In the DPCM, the prediction is subtracted from the actual pixel value to form a differential image that is much less correlated than the original image data. The differential image is then quantized and encoded. The quantization process determines the resulting bit rate and image quality (see figure 3).

3.3 Coding scheme used for compression

The coding scheme which is implemented in this communication, is applied on 256x256 grey-scale images corresponding to the resolution level 0, with each pixel taking one value from 0 to 255. In first, wavelet transform is performed to obtain the sub-images at the resolution level 1 (size 128x128). In second, VQ is directly applied to the details sub-images, and DPCM on the low resolution level 1 sub-image. We could have applied wavelet transform on the low resolution level 1 sub-image, but the in terms of compression, the result would have been the same.

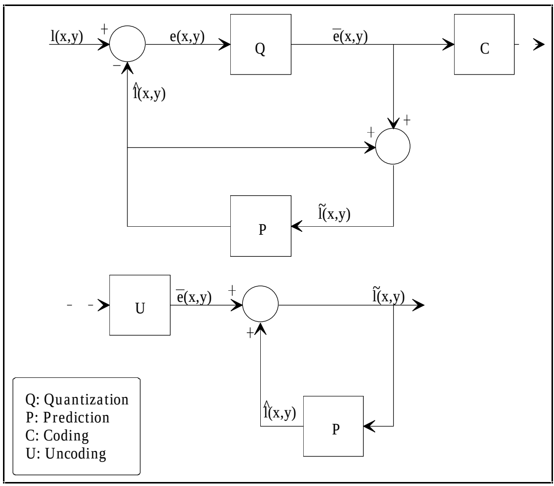

Figure 3 : DPCM scheme

4. IMPLEMENTATION ON A TRANSPUTER NETWORK WITH SYNDEX

Algorithm implementation with SynDEx, needs only the creation of a coherent GFDC. In first the data which condition the graph execution, need to be choosen. For an image processing application, a graph can be rhythmed by different types of data flow: image, line, or pixel. In this case we expected that the maximum of parallelism can be explained with lines.

4.1 Coding description

Wavelet decomposition is made with convolutions and sampling on images lines and columns. (see figure 4). These operations are realized in two steps for a 256 pixel line (see figure 5): (i) result of horizontal convolution in a sliding window; (ii) vertical convolution with the output sliding windows made every two lines. This method needs to conditione the output sliding window data flow, because the algorithm must compute some lines before the first vertical convolution was significant. The vector quantization is made when 4 lines are available. These 128 float lines are transformed into 32 matrices sizing 4x4 for explain the parallelism. The GFDC is presented by figure 6.

4.2 Uncoding description

The uncoding allows to reconstruct image with the low resolution sub-image coded and the details sub-images quantized. Convolutions and over-sampling are needed to compute the result (see figure 7). The first step of over-sampling columns doubles the data flow. After, two sliding windows are used for the vertical convolutions. Like for the coding, the output sliding windows data flow must be conditioned.

5 RESULTS AND CONCLUSION

By using this method to implement an image compression algorithm on processor network, we showed the parallelism of this application: it takes 80 seconds with 1 transputer for the coding a 256x256 image defined on 256 grey levels, and goes down to 25 seconds with 9 transputers; the decoding takes 25 seconds with one, and 15 seconds with 3, 4, 7 or 9 transputers. The decoding needs less operations than the coding; so the use of several transputers is not necessary.
This results are obtained with SynDEx 3.6. The next version 4.0 will be available for more processors than now (INMOS T800 and Texas Instrument TMS320C40). The C code generated will be also optimized to give better communication time performances.
Beyond of these results, we were determined on demonstrating the advantage of SynDEx for developing applications on multi-processors. This approach obviously reduces the time taken for development, and comes to optimized implementations, without bug in the communications.

6 REFERENCES

[1]: C.CUDEL, '*Parallélisation d'algorithmes de compression d'images sur un réseau de transputers*', thesis, University of Reims Champagne Ardenne, FRANCE, July 10th 1995, speciality Automatic and signal processing.

[2]: C.LAVARENNE, O.SEGHROUCHNI, Y.SOREL, M.SORINE, '*The SynDEx software environment for real-time distributed systems design and implementation*', Proc. of the European Control Conference (July 1991).

[3]: C.LAVARENNE, Y.SOREL, '*Optimisation et Generation d'Executifs Distribues Temps-Reel pour Algorithmes specifies avec les Langages Synchrones*', Proc.Conf. RTS94 (Paris, January 1994).

[4]: Y.MEYER, *'Wavelets and operators'*, in: Analyse at Urbana, I. (Eds.: Berkson, E. R.; N. T. Peck; J. Uhl) Cambridge Univ. Press: cambridge 1989, pp. 256-365.

[5]: S.G.Mallat, *'A theory for Multiresolution Signal Decomposition: The Wavelet Representation'*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, n°17, p.674-692, July 1989.

[6]:B.Ramamurthi, A. Gersho, *'Classified Vector Quantization of Images'*, IEEE Transactions on communications, Vol. COM-34, n° 11, November 1988.

[7]:M. Rabbani, P.W. Jones, *'Digital Image Compression techniques'*, SPIE Optical Engineering Press, 1991.

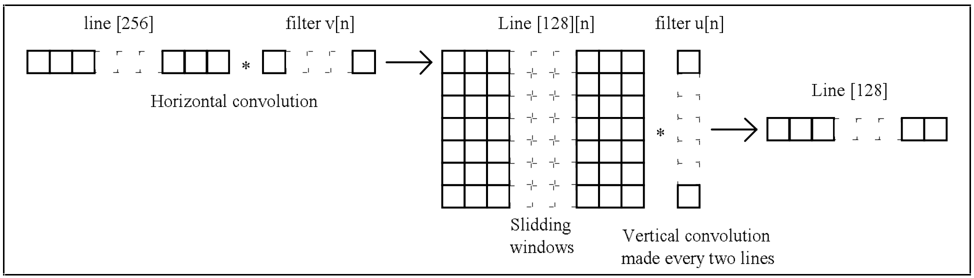

Figure 4: convolutions and sampling for wavelet transform decomposition

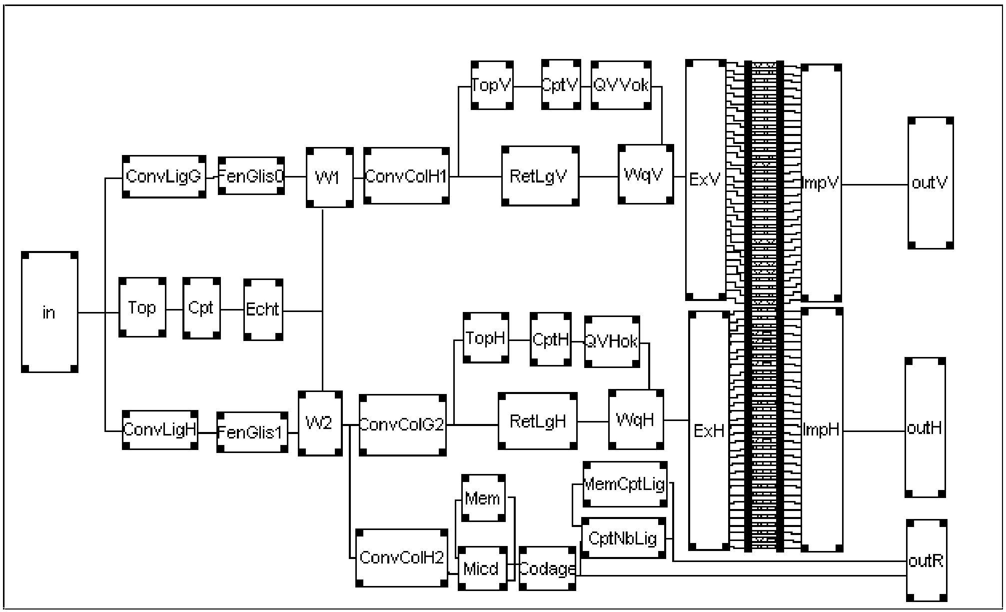

Figure 5: CDFG for 256x256 gray-scale images coding

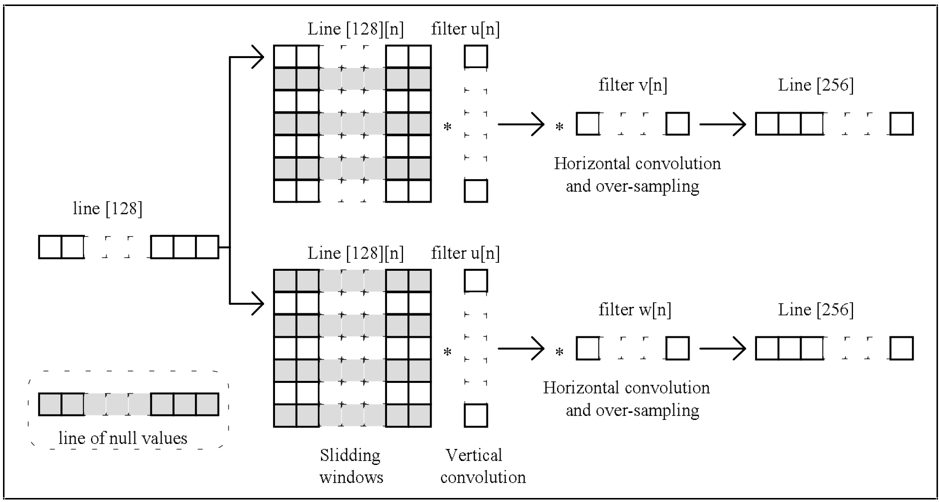

Figure 6: convolutions and over-sampling for wavelet transform reconstruction