Parallel Implementation of a 3-D Subband Decomposition Algorithm for Digital Image Sequence Compression on the Cray T3D

H. Nicolas, Cray Research, Switzerland, Lausanne, Switzerland; *M. Schutz*, and *F. Jordan*, Signal Processing Laboratory, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland

Abstract

This paper presents an efficient massively parallel implementation on the CRAY T3D of a digital image sequence compression scheme based on a 3-D subband decomposition. This compression method has been selected to be implemented on the CRAY T3D for its high potential of parallelization, its high computational complexity and its scientific interest. This implementation has been performed in C, using CRAY-SHMEM routines and PVM. The retained data partition has been chosen in order to minimize the communication, to optimize the load balancing and to minimize cache miss.

1 Introduction

In recent years, there has been a growing interest in digital image sequence data compression, related to the emergence of various applications such as High Definition Television (HDTV), digital television or videoconference. Effectively, due to the huge amount of data which should be transmitted (for example, an HDTV sequence represents around 660 Mbits/s), efficient compression algorithms should be used. To reach high compression ratio, these algorithms have to reduce as efficiently as possible both spatial and temporal redundancies existing in the image sequences.

Unfortunately, the computer requirements of image sequence coding algorithms are generally very high. This is due to the huge amount of data and to the fact that numerous operations are required per pixel. Parallel implementations will allow the optimization of some very relevant encoding parameters. Furthermore, more complex techniques can be investigated. Since the compression algorithms are generally highly parallelizable such kind of implementation is very promising [1].

With the advent of massively parallel systems such as CRAY T3D and the development of communication tools between the different processors such as PVM (Parallel Virtual Machine) and CRAY T3D communication routines SHMEM, the possibility of implementing efficiently coding algorithms is nowadays available. In the framework of the joint CRAY Research - EPFL PATP Project (Parallel Architectures Technology Program) we developed a parallel version of a 3D subband-based image sequence compression algorithm.

This paper is organized as follows. Section 2 describes the main characteristics of the proposed 3-D subband-based compression algorithm. The parallel implementation is presented in Section 3. Experimental results are provided in Section 4. Finally, Section 5 draws the conclusions.

2 Description of the proposed compression scheme

2.1 General presentation of the compression/decompression scheme

The current coding/decoding systems for digital television such as international standard H.261, MPEG 1

and MPEG 2 are based on Discrete Cosine Transformation (DCT). Nevertheless, this transformation does not exploit efficiently natural image redundancies because the image is segmented into blocks which are processed independently thus creating annoying blocking artifacts. In contrast, 2-D subband-based systems have been introduced to operate on the full input image such that these redundancies are efficiently exploited [2]. Thus, much more compact image representations are achieved. More recently, 3-D subband-based compression algorithms (two spatial dimensions and one temporal dimension) have been introduced in order to exploit the temporal redundancy [3][4]. The block diagram of the compression scheme which has been parallelized is described in Fig. 1. This coder performs as follows. The initial images are decomposed by the 3-D subband decomposition algorithm. Then, each subband is divided into small blocks of s_txs_xxs_y coefficients. Due to the non-uniform distribution of energy in different blocks, most of the blocks have zero or small energy. In order to transmit only the relevant information, the energy of each block is calculated. Then, only the blocks for which the energy is higher than a given threshold T are uniformly quantized and coded using an arithmetic coder. The selection information requires only 1 bit for each block, and represents a small portion of the total transmitted information. At the decoder, the arithmetic decoder followed by the inverse quantization and the 3-D subband synthesis permits to recover the decoded images. A more detailed presentation of this digital image sequence compression technique is available in [5].

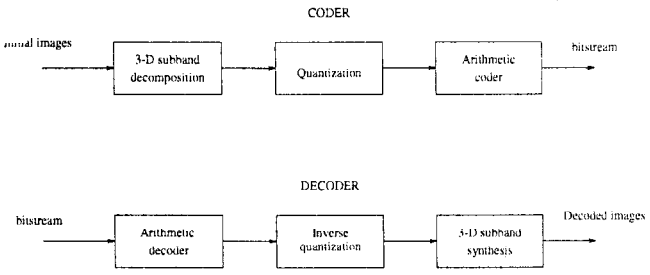

Figure 1: 3-D subband based compression scheme.

2.2 Description of the 3-D subband decomposition algorithm

The 3-D subband analysis decomposes digital video signals into various spatiotemporal frequency components of the 3-D time-frequency space. Each filtering process requires a convolution product for each pixel which implies a relevant computational cost. If $c(i)$ represents the i^{ieme} coefficient of the filter of length N then, for the one dimensional case, the discrete convolution product is given by:

$$Y(k) = \sum_{i=1}^{N} X(k+i-N/2-1).c(i) \qquad (1)$$

where Y is the filtered output and X the input signal. Since separate filter banks are used, the decomposition is successively performed in the horizontal (y), vertical (x) and temporal (t) directions (see Fig. 3). For each direction, a complete set of filters is applied in order to decompose the input image in its frequency bands, each filter being responsible for one band. A 2-band (high and low frequency) decomposition means that in each direction, two filters are used. A subsampling is introduced after each filtering in order to keep the number of samples constant (the subsampling factor is chosen equal to the number of filters in each filter bank). See also Fig. 2 which illustrates the subband decomposition in the 2-D case. The 3-D subband synthesis algorithms permit to recover the decoded images. For that, the transmitted subband coefficients are filtered successively in the t, x and y directions using the synthesis filter banks.

2.3 Evaluation of the complexity of the 3-D subband decomposition

If L_x, L_y and L_t denote the filter length in the x, y and t direction, respectively, then, a convolution product requires L_x (resp. L_y and L_t) multiplications and $L_x - 1$ additions (resp. $L_y - 1$ and $L_t - 1$). If $n.m.f$ (n rows, m columns and f images) represents the size of the 3-D data set, if S_x, S_y and S_t denote the subsampling factors in the x, y and t directions, respectively and if N_x, N_y and N_t denote the number of filters in the x, y and t direction, respectively, then, the total number C of operations needed to perform the 3-D subband decomposition is:

$$C = n.m.f.((L_x\frac{N_x}{S_x} + L_y\frac{N_y}{S_y} + L_t\frac{N_t}{S_t})mult.+ ((L_x-1)\frac{N_x}{S_x} + (L_y-1)\frac{N_y}{S_y} + (L_t-1)\frac{N_t}{S_t})add.).$$

For image compression applications, the subsampling factors are chosen equal to the number of filters of each filter bank, then,

$$C = n.m.f.((L_x + L_y + L_t) \text{ multiplications} + (L_x + L_y + L_t - 3) \text{ additions}).$$

In this paper, experiments are performed on CIF format image sequences (n=288, m=352, 25 images/sec.).

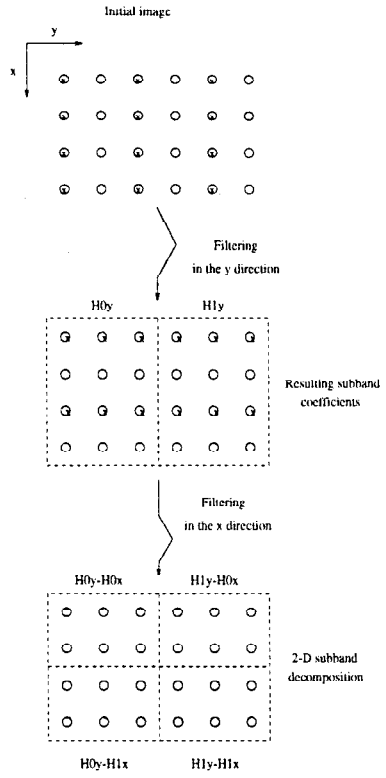

Figure 2: Illustration of a 2-D subband decomposition with 2 filters in the x and y directions. Samples for which the convolution must be done are marked with a "x".

The filter lengths are $L_x = L_y = 12$ for the spatial directions and $L_t = 2$ in the temporal one. In this case, the number of operations per second is:

$$C = (66\ 10^6 multiplications + 58\ 10^6 additions)/sec.$$

For digital television sequences (576x720), this cost (with the same filters) becomes:

$$C = (264\ 10^6 multiplications + 232\ 10^6 additions)/sec$$

3 Parallelization of the 3-D subband algorithm

The 3-D subband algorithm has been implemented in parallel on the CRAY T3D with 2^N Processing Elements (PE), $N = 0, ..., 8$. The initial images are first read by the processor 0. Then, these data are broadcasted to the local memory of all the other PEs using the fast *shmembroadcast* routine (Fig. 4 describes the I/O system). The 3-D data are then divided into N

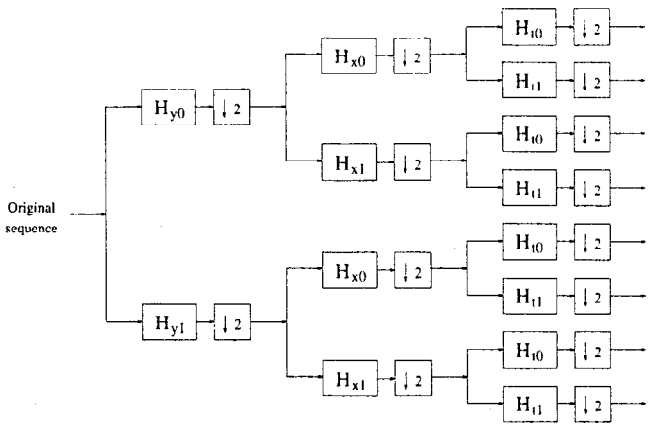

Figure 3: 3-D subband decomposition (with 2 filters in each direction). H0(x) (resp. H0(y)) and H1(x) (resp. H1(y)) represents the high and low pass filtering in the x direction (resp. y). T0 and T1 represent the high and low pass temporal filters. The subsampling factor is 2.

parallelepipeds. The decomposition is then performed in parallel for each processor according to the algorithm depicted in Fig. 1. The decoded images are finally written using the processor $n-1$. To be optimal, the data partionning must be performed according to the two following criteria:

1. Optimization of the load balancing.
2. Minimization of the communications.

The load balancing can be easily optimized if each processor computes the same number of data. This is done by dividing the initial 3D data set into same size parallelepipeds (see Fig. 5).
Since communications between processors has to be minimized in order to reach higher speed-up as possible, the length of the data partition boundaries has to be minimized. No communication is required for the decomposition in the y direction since all the initial images are available in the local memory of all PEs. In the x and t directions, communications are necessary at the boundaries of the parallelepipeds. In practice, no boundary (and therefore no communication) is introduced in the temporal direction because the amount of data in this direction is very low (typically 16 samples). In the x direction, communications can also be completly eliminated by using the data partition illustrated by Fig. 6. Nevertheless, when the number of PEs is high (or when the number of columns is low), the column number cannot be divided by the number of PEs. In this case, divisions are performed in the y

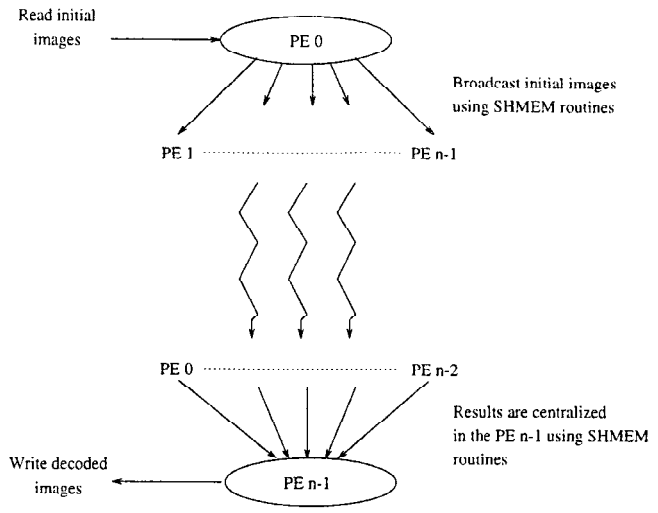

Figure 4: I/O system.

direction as much as possible to create the data partition and the remaining divisions must be performed in the x direction, thus creating communications.

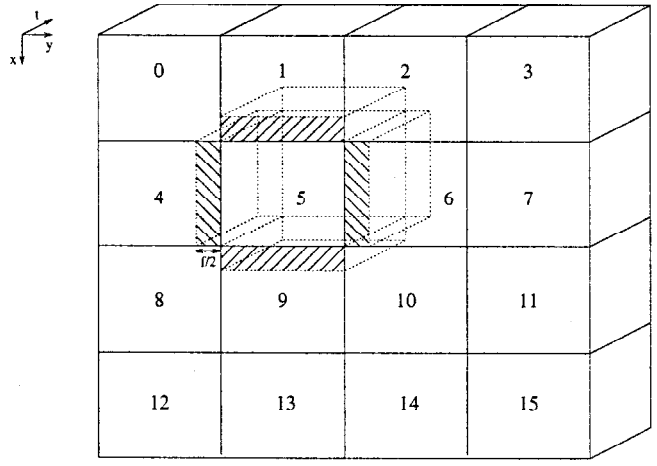

Figure 5: Data partitioning with 16 processors. f represents the lengh of the filters in the x and y directions.

4 Experimental results

This technique is applied to the test sequence "LTS" [6]. The format of this sequence is 288x352 pels/frame for luminance and 25 frames/sec. In the following experiments, two filters in each direction (one high pass filter and one low-pass) are used to decompose the images. The filter banks are identical in the horizontal and vertical directions. The length of the filters are 12 and 2 for the spatial and temporal directions respectively. The experiments have been performed using the data partition shown in Fig. 6. The size of the selection blocks is 32 ($s_t = 8, s_x = 2, s_y = 2$), and the number of frames is 16.

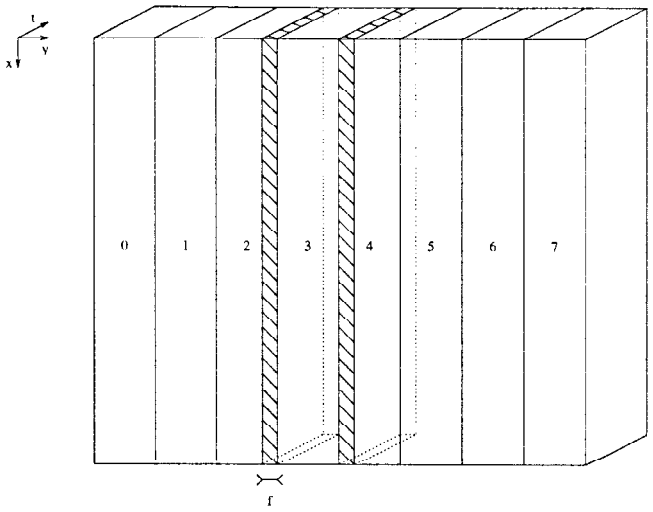

Figure 6: Data partitioning with 8 processors. f represents the lengh of the filters in the x and y directions.

Figure 7 shows the obtained speed-up for 1 to 128 processors. It can be seen that a speed-up of 108 is obtained using 128 processors for the 3-D subband decomposition, and a speed-up of 84 for the whole compression scheme. Furthermore, table 1 shows that for CIF format images (288*352 pixels, 25 Hz), the parallel algorithm is able to code the image sequences in real time for 32 processors and more. Theoretical speed-ups are not reached for the following reasons:

- As explained in Sec. 3, when the number of processors increases, boundaries are created in the x direction. Consequently, communications are needed. In the case of CIF format images, such communications appear for 32 processors and more. For 128 PEs, the communication time per processor is about 0.01 second for 16 frames (7.6 % of the total coding time).
- The number of bytes coded with the arithmetic coder depends on the original images. It is different from a processor to the other ones, therefore, the load balancing is not optimal. This problem is clearly illustrated by Fig. 7 which show the degradation of the speed-up when the arithmetic coder is taken into account.

Finally, single PE optimization allowed to reach 50 $.10^6$ operations per second per processor (17 $.10^6$

#PE	Compression time (second)	Number of compressed images per second
1	14.2	1.1
2	6.92	2.3
4	3.60	3.6
8	1.90	8.4
16	1.07	15.0
32	0.57	28.1
64	0.33	48.5
128	0.17	94.1

Table 1: Compression time (subband decomposition, quantization, arithmetic coder).

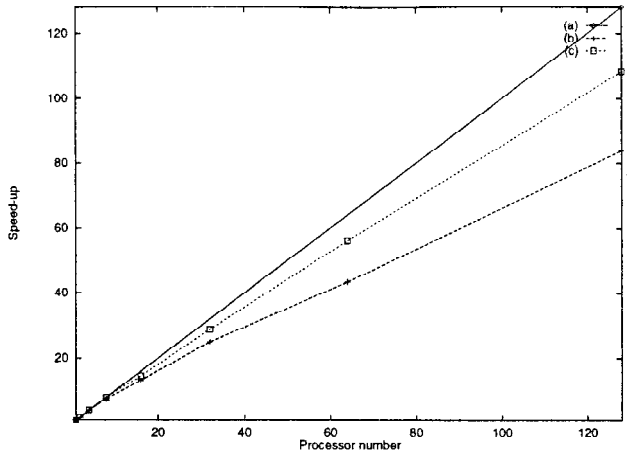

Figure 7: (a) Theoretical speed-up. (b) Speed-up for the whole compression scheme (subband decomposition, quantization and arithmetic coding). (c) Speed-up for the 3-D subband decomposition.

floating operations and 32 $.10^6$ integer operations per processor per second).

The parallel implementation permits an easy optimization of the algorithm parameters. Figure 8 shows the experimentals results which have been obtained in order to optimize the choice of the parameter T (energy threshold) and Q (quantization step). It shows the Pick signal to Noise Ratio (PSNR) versus the compression ratio for different value of T. For a given energy threshold, the compression ratio is increased by an increase of the quantization step Q (in Figure 8, Q increases from 1 to 30). The curves show that a value of $T = 5$ is optimal. Figures 9 and 10 show the compression ratio and the PSNR versus frame number, for $T = 5$ and $Q = 20$. This curve shows the stability of the results throughout the sequence. The compression ratio is equal to 10 with a PSNR of 24 dB for the 480 frames of the LTS sequence (19 seconds).

5 Conclusion

The parallel implementation on the CRAY T3D of image sequence coding algorithms is of major importance to perform the numerous experiments needed to evaluate seriously their performances and to permit an efficient optimization of the coding parameters. It should be pointed out that such experimental study cannot be easily performed using a sequential approach. The performance obtained demonstrates the interest of the massively parallel implementation since high speed-up (84 for 128 processors) have been obtained. Furthermore, for 32 processors and more, images are compressed in real time (for CIF format images). Using this parallel implementation, the two main parameters of the algorithm have been easily optimized in order to find the optimal trade-off between the quality of the decoded images and the compression ratio.

References

[1] H. Nicolas, A. Basso, E. Reusens and M. Schutz. Parallel implementations of image sequence coding algorithms on the CRAY T3D. *Supercomputing Review, No 6, EPFL, pp. 28–32*, November 1994.

[2] J.W. Woods. Subband image coding. *Kluwer Academic Publishers, Boston*, 1991.

[3] M. Vetterli. Multi-dimensional subband coding: some theory and algorithms. *Signal Processing*, Vol. 6(2), pp. 97-112, 1984.

[4] J.R. Ohm. Three dimensional subband coding of video. Vol. III, pp. 229-232, 1992.

[5] W. Li and M. Kunt. Block adaptive 3–D subband coding of image sequences. Vol. I, pp. 532-536, 1993.

[6] A. Nicoulin. The LTS/EPFL video sequence for very low bit rate coding. Technical Report 11, Swiss Federal Institute of Technology-LTS, September 1994.

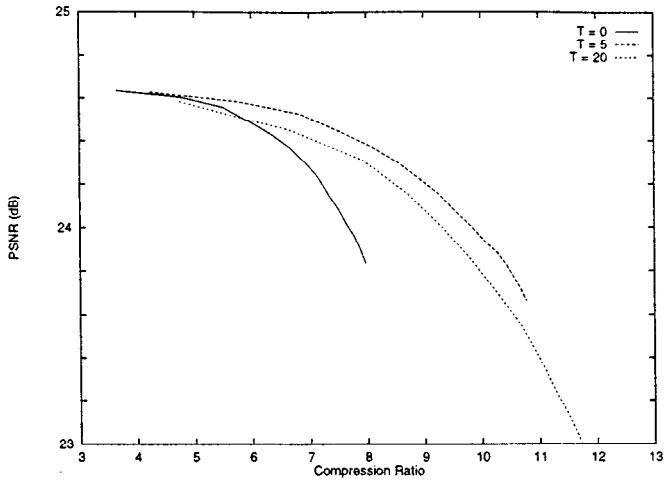

Figure 8: Compression ratio versus Pick Signal to Noise Ratio (PSNR) for different value of the energy threshold *T*.

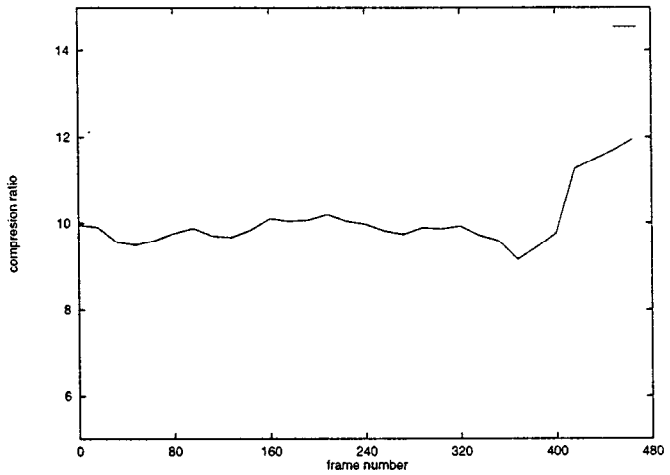

Figure 9: Compression ratio versus frame number for *T* = 5 and *Q* =20.

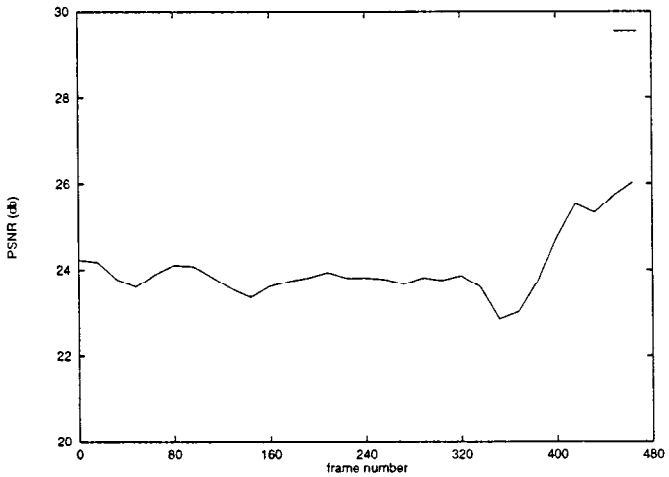

Figure 10: Pick Signal to Noise Ratio (PSNR) versus frame number for *T* =5 and *Q* 20.