

5

An Integrated Highly Parallel Architecture for Image Reconstruction *

Didier Lattard, and Guy Mazare

The large amount of information and computations is a critical problem in image processing. In this paper we show a highly parallel method to do image reconstruction which performs at real-time, using an asynchronous cellular array. The highly parallel architecture we propose is novel, its main characteristic is the message communication mechanism based upon message routing and mailbox principles.

After introducing the image reconstruction problem, we present the main reconstruction techniques and the sequential algorithms. We explain how to process these algorithms on a network. We describe this new integrated parallel architecture, its originalities and the system performing the whole reconstruction. We present the efficiency of this parallel image reconstruction method and the performance of the network.

1. Introduction

Image reconstruction is a tool to visualize the internal structure of an object or a living body. The process is used in many scientific fields including radiology, radio astronomy, electron microscopy, optics, holography and in industry. As part of radiology, that allows us to analyze the structure of human organs, detect eventual tumours and then help medical diagnosis (computed tomography, scanners). All these applications have been created a particular interest for image reconstruction problem. Measurements of radiation passing through an object at a certain level make it possible to calculate the internal distribution of density, visualized as a section image of the object. The reconstruction problem can be stated as follows: *estimate from a finite number of projections the density distribution in the cross-section of the original object* [12].

* This work is supported in part by the "Pôle Architecture" of the group "Coopération, Concurrence et Communication" (C3) of the Centre National de la Recherche Scientifique (CNRS).

2. Image Reconstruction: Problem and Methods

Problem

In this part we explain how the radiation measurements which permit the reconstruction are obtained and we formulate the problem.

For the acquisition of the projections we x-ray the object to study following the plane in which we want to realize the image section and we collect the rays after their passage through the object [1]. The set of data collected at each angular setting is called a projection. The ray intensity decreases with the quantity of material passed through, this ray attenuation is the line integral on its path of the object density:

$$\text{attenuation} = \int_{\text{path}} \text{density}$$

We want to reconstruct an image representing the cross-section. It is divided in N little squares (the pixels), a_i is the value of the i^{th} pixel, P_k the ray sum of the k^{th} ray (attenuation). We can write:

$$P_k = \sum_{i=1}^{i=N} w_{ik} \times a_i$$

where w_{ik} is a geometric factor (effect of the k^{th} ray on the i^{th} pixel. w_{ik} is null if the ray does not pass through the pixel) [2].

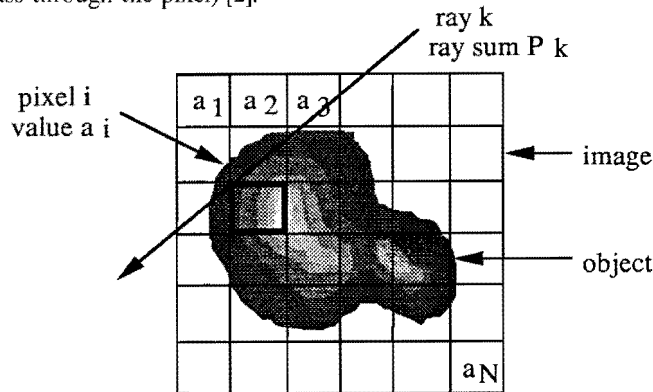


Figure 1: Image reconstruction: ray attenuation in different directions is used to reconstruct the original object density distribution.

The image reconstruction problem is to determine the pixel values a_1, a_2, \dots, a_N from the set of ray sums P_k obtained by radiating around the object. We have to resolve a linear system of K equations (number of radiation measurements P_k) with N unknowns (number of image pixels).

Reconstruction methods

There are two ways to classify reconstruction techniques. Firstly we can group them in two classes: analytic methods and algebraic methods. The former class is based on the integral relation between the ray sums and the pixel values, the latter class tend to solve the previous linear system. Secondly the reconstruction algorithms can be classified into four categories: summation methods, transformation methods, direct analytic methods and series expansion methods [4-6].

J. Radon worked on this subject and has shown that it is analytically possible to solve the problem [3]. His formulation is the basis for the current most commonly used image reconstruction technique: the Filtered Back Projection (FBP). This analytic method includes two parts: the filtering of the projections and the distribution of these filtered projections P_k to the pixels crossed by the ray.

Another efficient solution is to use an Algebraic Reconstruction Technique (ART). This implementation of the series expansion approach is an iterative process, which comes from the Kaczmarz's algorithm [7-11]. Each iteration of the ART method includes two phases. Firstly, for each ray, we uniformly distribute the ray sum P_k to the pixels crossed by the ray. Secondly, from the new pixel values, we calculate a set of pseudo ray sums P'_k . We repeat the process with the distribution of the differences $(P_k - P'_k)$ until the image variation is insignificant or until the calculated ray sums (P'_k) do not differ much from the measured ray sums (P_k). At each step, the image is refined and the final image (when $P_k \approx P'_k$ for all rays) is a reliable representation of the object density distribution in the studied plane [12].

Sequential algorithms

The commonly used methods are the Filtered Back Projection method and the Algebraic Reconstruction Technique. In the algorithmic descriptions, we use the primitives of the OCCAM language. SEQ is a sequential constructor, PAR is a parallel constructor, IF is a conditional constructor, WHILE and REPEAT UNTIL are repeated constructors. The Filtered Back Projection algorithm is:

```

SEQ
  initialization          ai := 0 for all pixels
  phase (0)              projection filtering
  phase (1)              back projection of the filtered measurements

```

The projections are acquired in a sequential manner. For each projection angle, the projection vector is filtered and back projected to the image by distributing the ray sums to the crossed pixels. The filtering is a one-dimensional computation. It can be performed by convolution or multiplication in the Fourier domain. The back projection (phase (1)) is a two-dimensional computation, all the pixel values are modified.

The Algebraic Reconstruction Technique algorithm is:

```

SEQ          /* initialization */
ai := 0 for all pixels
Pki := 0 for all rays

REPEAT      /* iteration step */
  SEQ
  phase (1) distribution of (Pk - Pki) on the ray path to modify the ai values
  phase (2) evaluation of a set of pseudo ray sums Pki using the previous ai values
  test      for each ray, check the difference between Pk value (= measured)
           and Pki value (= calculated)

UNTIL convergence (Pk ≈ Pki)

```

Phase (1) of both algorithms are identical. Phase (2) corresponds to a projection process. Both phases imply a large number of computations because they are two-dimensional.

It is evident that the Algebraic Reconstruction Technique is much more slow than the Filtered Back Projection method, because the projection process and the back projection process need the same amount of computation, whereas the algebraic technique need about twenty iterations. However, the iterative method is best in case the measurements are not complete or contain much noise.

3. Parallel Algorithm and Asynchronous Cellular Array

Parallelism

If we consider the computational requirements of the algorithms and the requirements on the image quality, we can easily see that a monoprocessor architecture cannot support a real time process (i.e. generate an image a few seconds after acquiring the radiation measurements). An easy way to evaluate these algorithms in a highly parallel manner is to use an array of processors, each one dealing with one pixel. The information about each ray forms a message, travelling through the network. A host computer manages the network inputs/outputs and synchronisation between the different computation phases.

Asynchronous cellular array

For some years, our team has been working on an original architecture: a highly parallel asynchronous cellular array [13]. The topology of the network is a 2-D grid with bidirectional connections to the four nearest neighbours (see Figure 2). Each cell is a processor able to realize simple operations. The main characteristic of the network is its message communication mechanism based upon mail-box principle which allows each cell to communicate with its four neighbours. For that reason, each cell is surrounded by eight unidirectional buffers (four input buffers and four output buffers). The cells are asynchronous: they compute independently and only when they have messages to process [14].

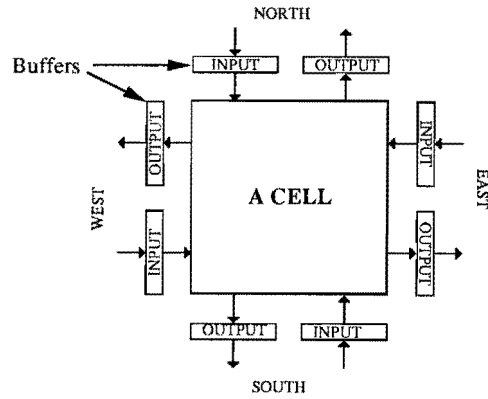


Figure 2: Detailed account of one cell.

The messages move between cells through the buffers. Each of them has a flag that indicates its state to the neighbouring cells: full (there is a message in the buffer) or empty (no message). A message contains many information fields:

- its type (projection, back projection, image input, image output)
- a ray sum field, who's content depends on the type. In case of back projection it is the ray sum distributed to the crossed cells, in case of projection it is the pseudo ray sum calculated by the crossed cells
- geometric information, that is the ray entry position in the cell and the ray angle (cosine, sine).

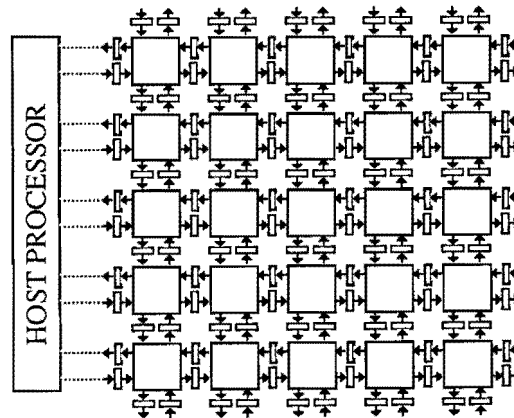


Figure 3: A 5×5 network connected to the host computer which manages it.

Host Computer algorithm

The host processor takes care of message sending and receiving and the iteration steps of the ART algorithm. The network allows to parallelize the two algorithm computation phases.

The host computer algorithm in the case of the two reconstruction methods is as follows:

Filtered Back Projection

```

SEQ
  initialize the network
  send the back projection messages
  receive the issued messages from the network
  receive the results (final image)

```

Algebraic Reconstruction Technique

```

SEQ
  initialize the network
  REPEAT
    SEQ
      PAR
        send the back projection messages
        receive the issued messages from the network
      PAR
        send the projection messages
        receive the issued messages from the network
        compare the calculated  $P'_k$  values and the measured  $P_k$  values
    UNTIL ( $P_k - P'_k$ ) sufficiently small
  receive the results (final image)

```

Image reconstruction machine

The objective of this work is to realize the system performing the whole reconstruction: the network and the host software. Figure 4 illustrates this system.

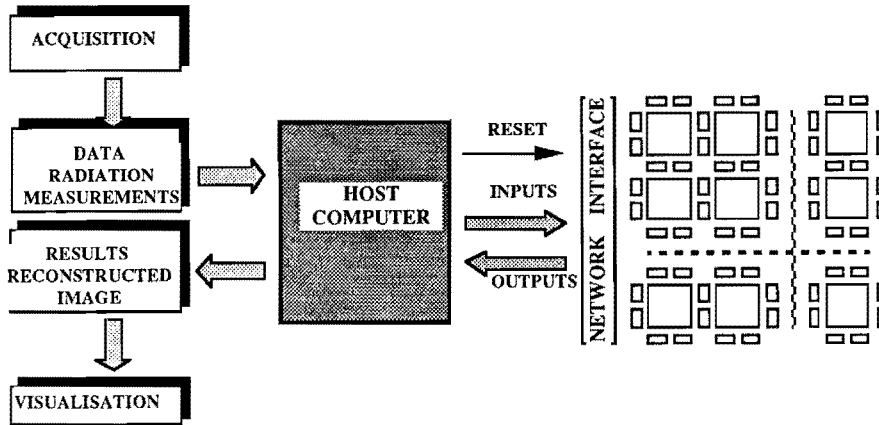


Figure 4: Image reconstruction machine

Cell algorithm

The principle of the algorithm, as performed by a cell is simple. After inspection of the input buffers to acquire an eventual message, the cell determines to which of its neighbouring cell it has to transmit the message. The transmission can be performed (addressee buffer is empty) the message is processed, otherwise the message is kept in the input buffer.

The cell algorithm is:

```

SEQ
  select an input buffer
  WHILE TRUE
    SEQ
      test the flag of the selected input buffer
      IF the selected input buffer is full
        SEQ
          read the message in the input buffer
          generate the next address
          test the flag of the addressee output buffer
          IF the addressee output buffer is empty
            SEQ
              empty the input buffer
              process the message
              write the message in the output buffer
              fill the output buffer
          select the next input buffer
    
```

Cell computations

The most complex parts to implement are computation of the next address and processing of the message. Before detailing these parts, we define the geometric information of the message in Figure 5.

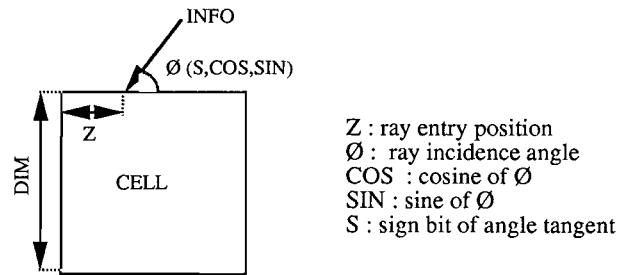


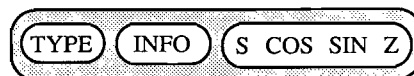
Figure 5: Geometric information send in each message.

The computation of the next address has several objectives:

- determine the addressee cell (it is necessary to know what cell the message will be passed before processing it, because a cell will process the message only if it can be routed),
- compute the new geometric information of the message that eventually will be passed after processing,
- compute the length of the ray path across the pixel (this result will be used when processing the message. It is the geometric factor w_{ik}) mentioned before.

Processing the message depends on its type. For the back projection method, the cell adds the product (ray sum value \times length of the path across the pixel) to the pixel value. For the projection method, the cell adds the product (pixel value \times length of the path) to the pseudo ray sum value received.

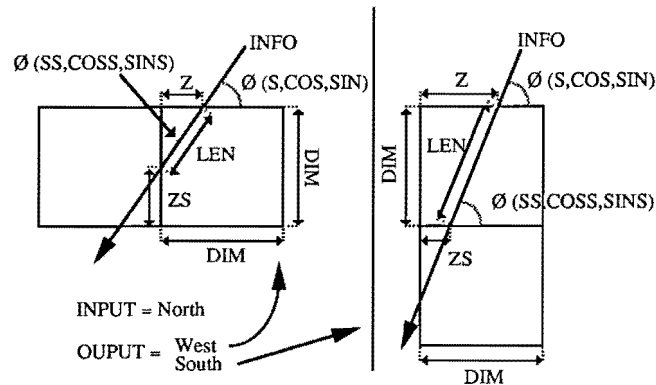
The message structure is:



For computation of the next address, we have to determine the geometric information of the next message $COSS$, $SINS$, SS , ZS , the output side of the ray $OUTPUT$ and the length of the path LEN . All these calculations are function of the geometric information of the message received and of the input side of the ray $INPUT$.

We can detail the equations and illustrate them in case $S=0$ (tangent $TG = \text{SIN}/\text{COS} > 0$):

| | |
|---|--|
| $\text{LEN} = Z/\text{COS}$ | $\text{LEN} = \text{DIM}/\text{SIN}$ |
| $\text{ZS} = \text{DIM} - (Z \times \text{TG})$ | $\text{ZS} = Z - (\text{DIM}/\text{TG})$ |
| $\text{COSS} = \text{SIN}$ | $\text{COSS} = \text{COS}$ |
| $\text{SINS} = \text{COS}$ | $\text{SINS} = \text{SIN}$ |
| $\text{SS} = 1 - S$ (binary complement) | $\text{SS} = S$ |



We can also pass the angle tangent instead of cosine and sine and do an approximated computation of the length. This would make the messages shorter. This is a possibility because we do not necessarily need an exact computation of the length.

Efficiency

The parallel method efficiency with respect to the sequential method is evident. All pixels are processed simultaneously, the cost saved is in the order of n^2 ($N = n^2$: number of pixels). The computational costs of the sum of all cells is in the order of NK , where $K = Mm$, K is the total number of projected points, M is the number of projections and m the number of points of the projection. In practice there are approximately n points in a projection ($m \approx n$). Then the global amount of computation is in the order of n^3 by projection. For the cellular array that processes in parallel, the total cost is in the order of n per projection. The number of projections necessary to obtain a sufficiently good image is in the order of n ($M \approx n$).

Simulation

Functional simulations of a 32×32 network and host processor have been done to validate the algorithmic studies and the original cellular architecture. A simple language OCCAM based on CSP, allows the description of such concurrent processes and the PC-board B004 including a Transputer and two megabytes of memory is an easy and efficient way to study such parallel architectures [15, 16].

The results allow to verify the mathematic relation existing between the data coding and the image definition. The number of bits necessary to code information is in the order of $\log_2(n)$ (i.e. a logarithm with basis 2), where n^2 is the number of pixels of the image. In particular, if we want to reconstruct an 1024×1024 image, we can use 16-bit coded integers.

We have done a temporal simulation using SILVAR-LISCO software to study the network activity and the message progress inside the network. After a first study of the architecture and its timing and taking into account the real-time constraints (acquisition speed of the projections: about 10 msec. per projection vector), we found that a cell processes rather quickly (the time processing for 1 pixel is about $5 \mu\text{sec.}$) due to which the network use is not optimised.

Given this, the idea arose to associate an array of pixels to one processor. The messages can have the same structure and each cell performs an iterative process to deal with its pixels array. The number of pixels a cell has to process, must be optimized. If it is too small, the network is not optimally used because the messages pass through the network very rapidly and only a few cells process simultaneously. If it is too large, the first messages will block the network and the other messages have to wait before being processed.

If a cell corresponds to an array of $k \times k$ pixels, we found that k^2 should be in the order of the ratio of the projection acquisition time to the pixel computing time.

Cell architecture

The cell architecture (see Figure 6) is based on a Programming Logic Array for the control part, an operative part including registers, an Arithmetic Logic Unit (performing addition and subtraction) and a little Random Access Memory to store pixel values. A peripheral bus surrounds the cell body. It allows to access the buffers and is also used by the operative part as data bus. Data are 16-bit integers.

4. Integration

Now that we know what performance such a technique of parallel image reconstruction can give, we set our goal to integrate a maximum number of elementary cells on a single chip, with the perspective of working with a high resolution image. Existing systems deal with images that have a maximum of 1024×1024 points. Of course, we could not think of integrating such a network on a chip today, but future advances may make this possible. We can follow two possible directions to oppose this limitation: develop a board with an array of chips, each one realizing a small size network, or take interest in the Wafer Scale Integration aspect, that today becomes more and more realistic [17, 18].

The primary goal is to realize a chip including (4×4) cells, each one dealing with a (16×16) pixels array, to process a (64×64) image.

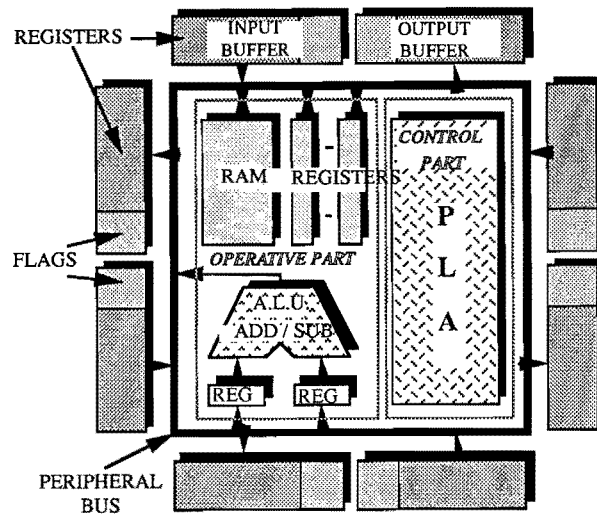


Figure 6: The architecture of the cell.

Up to now, this particular architecture has been designed and evaluated by using simulation techniques. However, most of this work is based upon ideas and results that are developed by the whole team working on asynchronous cellular arrays. The whole circuit has been designed and processed, in a CMOS 2μ , 1 metal technology, which includes 2×2 cells and interfaces, but is devoted to an other application (logical simulation). These experiments had led to the design of a cell library, including message-based communication and operative part architecture, which will be of interest for this particular application (image reconstruction). Moreover, most of the simulations have been based upon basic message communication delay and cycle times coming from the circuit already realized. The realization of the specific IC will take advantage of all these previously made parts.

5. References

- [1] R.A. Robb and C.R.C. Crit., "X-ray computed tomography: an engineering synthesis of multiscientific principles", *Reviews in Biomed. Eng.*, vol. 7, 1982, pp. 265-333.
- [2] N. Kingswood, E.L. Dagless, R.M. Belchamber, D. Betteridge, T. Lilley and J.D.M. Roberts, "Image reconstruction using the Transputer", *IEEE Proceedings*, vol. 133, part E, no. 3, 1986, pp. 139-144.
- [3] J. Radon, "Über die Bestimmung von Funktionen durch ihre Integralwerte langs gewisser Mannigfaltigkeiten", *Berl. Verh. Sachs. Akad. Wiss. Leipzig, Math-Nature KI.* 69, 1917.

- [4] G.T. Herman, "Image reconstruction from projections: the fundamentals of computerized tomography", Academic Press, New York, NY 1980.
- [5] G.T. Herman and F. Natterer, "Mathematical aspects of computerized tomography", Springer Verlag, Berlin, Germany, 1980.
- [6] R.M. Lewitt, "Reconstruction algorithms: transform methods", *Proc. IEEE*, vol. 71, no. 3, 1983, pp. 98-116.
- [7] R. Gordon, G.T. Herman and S.A. Johnson, "Image reconstruction from projections", *Scientific American*, vol. 233, no. 4, 1975, pp. 56-68.
- [8] R. Gordon, R. Bender and G.T. Herman, "Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography", *Journal of Theoretical Biology*, vol. 29, 1970, pp.471-481.
- [9] R. Gordon, "A tutorial on ART (Algebraic Reconstruction Techniques)", *IEEE Trans. Nucl. Sci.*, NS-21, 1974, pp. 78-93, 95.
- [10] Y. Censor, "Finite series-expansion reconstruction methods", *IEEE Proceedings*, vol. 71, no. 3, 1983, pp. 409-419.
- [11] G.T. Herman and A. Lent, "Iterative reconstruction algorithms", *Comput. Biol. Med.*, vol. 6, 1976, pp. 273-294.
- [12] D. Lattard and G. Mazare, "Parallel image reconstruction by using a dedicated asynchronous array", *Parallel Processing for Computer Vision and Display*, Leeds, 1988.
- [13] Y. Ansade, R. Cornu-Emieux, B. Faure, G. Mazare and P. Objois, "Algorithms dedicated to a network of asynchronous cells", in *Parallel Algorithms and Architectures*, M. Cosnard & Y. Robert Ed., North Holland, 1986.
- [14] R. Cornu-Emieux, G. Mazare and P. Objois, "An Integrated Highly Parallel Architecture to Accelerate Logical Simulation", *ISELDECS 87*, Kharagpur, INDIA
- [15] C.A.R. Hoare, "Communicating Sequential Processes", *Communication ACM*, vol. 21, no. 8, 1978, pp. 666-677.
- [16] "OCCAM Programming Manual", INMOS Limited, Prentice/Hall, 1984.
- [17] T. Leighton and C.E. Leiserson, "Wafer Scale Integration of systolic arrays", *IEEE Transactions on Computers*, vol. C34, no. 5, 1985.
- [18] Y. Ansade, R. Cornu-Emieux, B. Faure and G. Mazare, "WSI asynchronous cell network, in Wafer Scale Integration", G. Saucier, L. Trihle Ed., North Holland 1986.