IAPR Workshop on CV - Special Hardware and Industrial Applications OCT.12-14, 1988, Tokyo

# AN INTEGRATED ASYNCHRONOUS CELLULAR ARRAY TO DO PARALLEL IMAGE RECONSTRUCTION

Didier LATTARD, Guy MAZARE
Laboratoire de Génie Informatique
Institut IMAG

c/o USSI-INPG
46, avenue Félix Viallet
38031 Grenoble Cedex
France

## ABSTRACT

We show in this paper how it is possible to reconstruct an image, in a very parallel way, using an asynchronous cellular array, with real-time performance. After introducing the image reconstruction problem, we present the main reconstruction techniques and the sequential algorithms. Then, we explain the manner 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.

## INTRODUCTION

The image reconstruction is a way 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 ... As part of radiology, this allows us to analyze the structure of human organs, detect eventual tumours, and then help medical diagnosis (computed tomography).

Measurements of radiation passing through an object at a certain level, make it possible to calculate its internal distribution of density : 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.*

## IMAGE RECONSTRUCTION PROBLEM

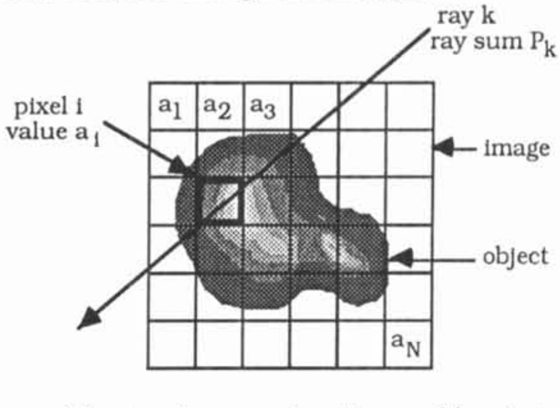
We x-ray the object to study, following the plane in which we want to realize the image section; we collect the rays after their passage through the object [1]. The set of data collected at each angular setting is called a projection. The ray intensity decreases with the quantity of material passed through,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

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

The cross-section 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 :

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

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].



The image reconstruction problem is to determine the pixel values $a_1, a_2, ..., a_N$ 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).

---

*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).*

## RECONSTRUCTION METHODS

We can classify the reconstruction algorithms into four categories : summation methods, transform methods, direct analytic methods and series expansion methods [4], [5], [6].

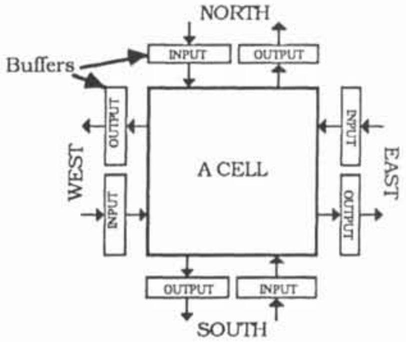
J. Radon worked on this subject, and has shown that it is analytically possible to solve the problem [3]. His formulation is the basis for the current most commonly used image reconstruction technique : the filtered back projection (FBP). This analytic method includes two parts : the filtering of the 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 a iterative process, which comes from the Kaczmarz's algorithm [7], [8], [9], [10], [11]. Each iteration of the ART method includes two phases : at first, for each ray, we distribute uniformly 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 calculated ray sums $(P'_k)$ are not very different of the measured ray sums $(P_k)$. At each step, the image becomes better, and the final image is a reliable representation of the object density distribution in the studied plane (when $P_k \approx P'_k$ for all rays) [12].

## PARALLEL ALGORITHM AND ASYNCHRONOUS CELLULAR ARRAY

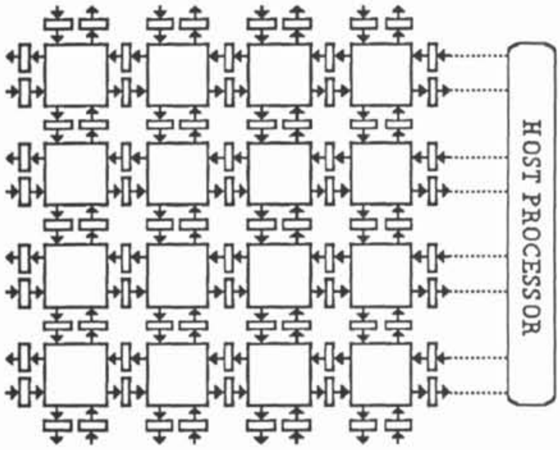
If we consider the large number of computation the algorithms have to perform, and a definition of the image we want to reconstruct good enough, we can easily see that a monoprocessor architecture cannot perform a real-time process (obtain an image a few seconds after acquiring the radiation measurements). An easy way to compute these algorithms in a very parallel manner is to use an array of processors, each one dealing with one pixel. The information about each ray form a message, moving into the network. A host computer manages the network inputs/outputs, and the synchronisations between the different computation phases.

Since a few years, our team is working on an original architecture : a highly parallel asynchronous cellular array [13]. The main specifications of the network are : each cell is a processor able to realize simple operations, each cell is surrounded with eight unidirectional buffers (four input buffers and four output buffers), each cell communicates with its four neighbours using a message routing mechanism, the cells are asynchronous, they compute independently and only when they have messages to process [14].



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 (back projection or projection), a ray sum field that differs with the type : ray sum distributed to the crossed cells (back projection), pseudo ray sum calculated by the crossed cells (projection), geometric information : ray entry position in the cell, and ray angle (cosine, sine).

The host processor realizes the messages sendings and receivings, and the repeated process of the ART algorithm; the network allows to parallelize the two algorithm computation phases.

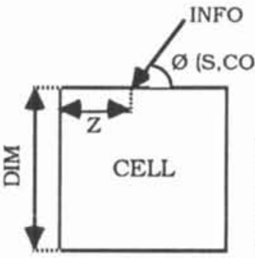


The principle of the algorithm, performed by a cell is simple. After inspection of the input buffers to acquire an eventual message, the cell determines its neighbouring cell to which it has to transmit the message, and if the transmission can be performed (addressee buffer is empty) then it processes the message, otherwise it keeps the message 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 to the output buffer
              fill the output buffer
      select the next input buffer
```

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



Z : ray entry position
Ø : ray incidence angle
COS : cosine of Ø
SIN : sine of Ø
S : sign bit of angle tangent

The computation of the next address has several objectives: determine the addressee cell, compute the new geometric information of the message, and the length of the ray path across the pixel (this result will be used in the process of the message, it is the previous mentioned geometric factor $w_{ik}$).

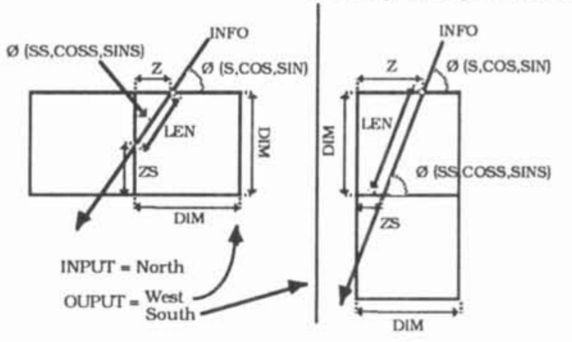
The process of the message depends on its type; for the back projection, the cell adds the product (ray sum value * length of the path across the pixel) to the previous evaluated pixel value. For the projection, the cell adds the product (pixel value * length of the path) to the received pseudo ray sum value.

The message structure is :

For the 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 received message, and of the input side of the ray INPUT. We can detail the equations and illustrate them in the case S=0 (tangent TG = SIN/COS > 0) :

LEN = Z/COS
ZS = DIM-(Z*TG)
COSS = SIN
SINS = COS
SS = S

LEN = DIM/SIN
ZS = Z-(DIM/TG)
COSS = COS
SINS = SIN
SS = 1-S
(binary complement)



We can also pass the angle tangent instead of cosine and sine, and do an approximate computing for the length, then the message is shorter. This is possible because we don't need a good precision for the length computing.

After a first study of the architecture and its timing, and taking account of the real-time constraints (acquisition speed of the projections : about 10 msec. by projection vector), we can say that the cell process is too fast (the time processing for 1 pixel is about 5 µsec.), and the network use is not optimised.

Then, a new idea is to associate an array of pixels to a processor. The messages have the same structure, and each cell performs an iterative process to deal with its pixels array.

The cell architecture is based on a Programming Logic Array for the control part, and an operative part including registers, a Arithmetic Logic Unit (performing addition and substraction) and a little Random Access Memory for storing pixel values. Data are 16 bit integer.

## SIMULATION

Functional simulations of the whole 32*32 network and host processor have been done to validate the algorithmic studies. 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].

We have made temporal simulation to study the network activity, and the message progress inside the network.

## EFFICIENCY AND INTEGRATION

The parallel method efficiency with respect to the sequential method is evident. All the pixels are processed simultaneously, the cost saved is in the order of $n^2$ ($n^2$: number of pixels). The global amount of computation of the sum of all cells is in the order of $n^3$ by projection; as for the cellular array that processes in a parallel manner, the total cost is in the order of n by projection.

When we know the performance of such a technique of parallel image reconstruction, the goal is to integrate a maximum number of elementary cells on a single chip, with the perspective of working with a high resolution image. The 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 todays though 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 is today more and more important [17], [18].

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

## REALIZATION

Up to now, this particular architecture has been designed and evaluated by using simulation techniques. But most of this work is based upon ideas and results that are developped by the whole team working on the asynchronous cellular arrays. Whole circuits has been designed and processed, in a CMOS 2μ, 1 metal technology, which include 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 simulations have been based upon basic message communication delay or cycle time coming from the realized circuit. And the realization of the specific IC will take advantage of all these previously made parts.

## REFERENCES

[1] *X-ray computed tomography : an engineering synthesis of multiscientific principles*, R. A. Robb, CRC Crit. Reviews in Biomed. Eng., Vol. 7, 1982, pp. 265-333.

[2] *Image reconstruction using the Transputer*, N. Kingswood, E.L. Dagless, R.M. Belchamber, D. Betteridge,T. Lilley, J.D.M. Roberts, IEEE Proceedings, Vol. 133, Pt. E, N°3, 1986, pp. 139-144.

[3] *Uber die Bestimmung von Funktionen durch ihre Integralwerte langs gewisser Mannigfaltigheiten*, J. Radon, Berl. Verh. Sachs. Akad. Wiss. Leipzip, Math-Nature Kl. 69, 1917.

[4] *Image reconstruction from projections : the fundamentals of computerized tomography*, G.T. Herman, Academic Press, New York, NY 1980.

[5] *Mathematical aspects of computerized tomography*, G.T. Herman, F. Natterer, Springer Verlag, Berlin, Germany, 1980.

[6] *Reconstruction algorithms : transform methods*, R. M. Lewitt, Proc. IEEE, Vol. 71, N°3, 1983, pp. 98-116.

[7] *Image reconstruction from projections*, R. Gordon, G.T. Herman, S.A. Johnson, Sci. Amer., Vol. 233, N°4, 1975, pp. 56-68.

[8] *Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography*, R. Gordon, R. Bender, G.T. Herman, Journal of Theorical Biology, Vol. 29, 1970, pp. 471-481.

[9] *A tutorial on ART (Algebraic Reconstruction Techniques)*, R. Gordon, IEEE Trans. Nucl. Sci., NS-21, 1974, pp. 78-93, 95.

[10] *Finite series-expansion reconstruction methods*, Y. Censor, IEEE Proceedings, Vol. 71, N°3, 1983, pp. 409-419.

[11] *Iterative reconstruction algorithms*, G. T. Herman, A. Lent, Comput. Biol. Med., Vol. 6, 1976, pp. 273-294.

[12] *Parallel image reconstruction by using a dedicated asynchronous array*, D. Lattard, G. Mazare, Parallel Processing for Computer Vision and Display, Leeds, 1988.

[13] *Algorithms dedicated to a network of asynchronous cells*, Y. Ansade, R. Cornu-Emieux, B. Faure, G. Mazare, P. Objois, in Parallel Algorithms and Architectures, M. Cosnard & Y. Robert Ed., North Holland, 1986.

[14] *An Integrated Highly Parallel Architecture to Accelerate Logicial Simulation*, R. Cornu-Emieux, G. Mazare, P. Objois, ISELDECS 87, Kharagpur, INDIA

[15] *Communicating Sequential Processes*, C.A.R. Hoare, Communication ACM, Vol. 21, N°8, 1978, pp. 666-677.

[16] *OCCAM Programming Manual* INMOS Limited, Prentice/Hall, 1984.

[17] *Wafer Scale Integration of systolic arrays*, T. Leighton, C.E. Leiserson, IEEE Transactions on Computers, Vol. C34, N°5, 1985.

[18] *WSI asynchronous cell network*, Y. Ansade, R. Cornu-Emieux, B. Faure, G. Mazare, in Wafer Scale Integration, G. Saucier, L. Trihle Ed., North Holland 1986.