# A 60X60 PROCESSOR ARRAY SMART SENSOR FOR BINARY IMAGE PROCESSING.

A.Reichart*, P.Garda**, E.Belhaire**, F.Devos**, B.Zavidovique***

* ETCA/CREA/SP Fort de Montrouge 94114 ARCUEIL CEDEX FRANCE
**IEF (CNRS LA 22) Bat 220 Université de Paris Sud 91405 ORSAY CEDEX FRANCE
*** IEF-ETCA

## Abstract.

Smart sensors could be useful in a lot of applications for a speedy and rough vision. We describe in this paper an architecture of a SIMD mesh array optical sensor, we give an implementation in this architecture of the main loop of an algorithm of pseudo-euclidian skeletonization and we present an example of control programmation, a counter of binary pixels made with elementary projections.

## Introduction.

Whereas computer vision research focuses on grey-level picture processing, a large number of industrial applications still use binary pictures [YOTE88]. As a very high processing speed is required, specialized architectures have been studied for a while [Reeves84]. Thanks to VLSI, high performance compact machines are now possible [Person88]. In this paper, we describe a monolithic parallel binary picture processing, usable as a smart sensor.

## Retina principles.

Efficient algorihms on binary pictures have been designed for a number of situations :

- preprocessing (noise filtering,...) and template matching (small templates detection, endpoints marking, objects minimum width check...)
- propagation (holes filling, single connected component extraction,...) and morphological features extraction (skeletons, ...)
- geometrical features measurement (area, perimeter, connectivity number,..) and morphological features measurement (granulometries,..)

They are followed by decision techniques (structural recognition, statistical classification, graph matching [Baker88],...)

These algorithms can be built out of iterations of a small number of primitive operations :

- unary local operator, in the sense of [Klette80], which compute a result picture out of a given picture through local boolean computations ;
- pointwise boolean operators between several pictures ;
- picture area measurement.

We have designed a parallel processor performing most of these algorithms while fitting in a single chip and including optical sensors. Its primitive operations are pointwise boolean operations and 2-D shifting in four directions (North, East, West, South). These primitive operations, when iterated and cascaded, provide most of the previous algorithms. By following this implementation, local operators of arbitrary size and shape defined by templates including "don't care" sites are easily achievable in a time linear with templates size and number.

Some examples of grey level picture processing are achievable with the optical sensor. A description will appear in [GRZ88].

## Retina architecture.

The idea of a cellular machine including optical input through photodiodes comes back to UCPR1 by M.Duff [Duff 67]. The successive evolutions of CLIP (number 1, 2, 3) led to the well-known CLIP4 array [Duff 86] which is well suited to both logical operations on binary pictures and numerical operations on multilevel pictures. Other cellular arrays have been built for greylevel picture processing, such as DAP [Reddaway 73]and MPP. Finally the GAPP can be considered as a buiding block for large mesh arrays.

Now as far as smart sensors are concerned, a monolithic realisation is required. But while increased densities bear more computing devices inside the chip, I/O pads do not multiply accordingly. This leads naturally to the conclusion that a large number of PEs need to be monolitically arranged, and in this case, to include an optical input results in a massive transfer of information between sensor and processor.

The retina architecture results from the class of operators it has to implement. It has been sketched in [Garda 85]. Fully parallel neighborhood combinatorial logic (NCL) is naturally operated through a SIMD mesh array. For the PE to be as small as possible, we have chosen to provide a fully sequential PE ; indeed even the communications of each PE with its neighbors are serialized. This results into a cost effective design, when only one way neighborhood operations, performed one at a time, are required. A NCL operator is determined through a support V and a sequence of binary templates $(Ti)_j$, each being a matrix of 0, 1 and "don't care".

To apply this cellular operator to a binary picture I is then a mere sequential comparison to each of these templates. Moreover, the comparison to a single template can be performed serially on the template elements, and in parallel by all PEs. For that purpose, the PE at site z has to perform serial comparison of the current template with its neighborhood (z + V). For each template site p, the PE at site z has to compare the I pixel value at site (z+p) with the pixel template at site p which is either a one, a zero or a "dont'care". The template pixel value at site p thus needs to be broadcast to all PE, and each PE needs a way to access the I pixel value at site (z + p) and has to compare it to the received template pixel value : that can be performed by an exclusive-or with the complement of the template pixel value. Then, the PE has to check whether all its neighborhood values match the corresponding template value, so it needs to perform the "and" of the result of all these comparisons. Finally, each PE has to check over a number of templates, and to check whether any of these matches its neighborhood. This can be done by "oring" the results of the comparison with each of the templates.

In order to perform these comparisons, the PE at site z has to access the pixels in its neighborhood (z+V).

There are no restrictions on V, and thus each PE has to access to any I pixel value. Fortunately all PE will access pixel values with the same offset.This access is serial, as the PE performs all computations serially. Moreover, accessing to a neighboring pixel is equivalent to translating the picture relatively to the PE array. Any translation can be performed by iteration of a single step translation. Thus the access to neighboring PEs can be performed by iteration of single step translations of the picture. These single steps can be performed thanks to a 2-D bidirectionnal shift register, which is built out of the interconnection of two 1-D bidirectional shift registers, as described for example in [Mead 81].

Of course, this architecture is simpler than those already suggested for processor arrays. The serial access to neighbors through shift registers differs from the gated input of CLIP4 PE, the multiplexed input of MPP PE and the NS/EW registers of GAPP PE. On the one hand, the combinatorial propagation capability of CLIP4 is lost. On the other hand, this implementation is particularly cost effective as it provides any translation through clever use of what is actually a memory hardware : any neighborhood can be realized that way.

The architecture features a SIMD mesh-connected processor array [Garda86]. The integrated circuit layout follows the logical design. In NMOS the PE size is less than 25 transistors and in static CMOS, it is less than 40. The size and shape of the PE layout is mainly determined by the control clocks number and thus depends on the interconnection layer number provided by the technology. With the current CMOS 2 metal 2 micron PE layout a 60x 60 fits in a 6.8 by 6.5 mm chip. We have realized a first temptative 8x10 retina, which has been fully tested (electronically and functionnally)

## Retina as a front-end for pattern recognition.

In this paragraph, the approach is similar to the standard approach to pattern recognition which relies on an image processing stage. Binary pictures have been dealt with for a long time, and numbers of suitable operators can be implemented. As it is, the retina can perform combinatorial cellular logic operations. These include erosions, dilations, and their iterations as opening, closing, ... Also operations relying on template matching are easily implemented : these include primarily binary edge detection, shrinking and thinning, whose implementation is as follows.

Iterative content :

Store in P2 the points where the templates are matched - this is a NCL-.

Reset P1 and transfer a copy of P in P1.

Copy P2 in P.

And P complement with P1 content.

Transfer P1 to P.

End of iteration content.

This process is then iterated until the whole thinning process is completed. Another useful primitive is binary propagation [Duff 86]. But it turns out that such a computation requires a supplementary memory point.

Moreover the addition of this extra memory point allows the implementation of a number of other algorithms. These are only examples which would better be fully and systematically investigated considered a precisely designed device. But the power of a full preprocessing stage for binary pictures towards statistical pattern recognition could be reached thanks to a global counter. It allows the computation of the area of a picture, and thus combining geometric operators with counting yields the full range of numerical features as area, intercept number, connectivity number, and also various histograms and granulometries. While [Reeves 80] defends the idea of an additionnal global counter, we show in the next section how to perform a counter in the smart sensor itself.

Then such a device can be connected to another chip which is devoted to classification as the one described by [Jain84]. At end a compact and fast pattern recognition device is built out of a very small number of chips.

## An NCL pseudo-euclidian skeletonization.

A local operation as the pseudo-euclidian skeletonization may be done with the smart sensor. In the algorithm described in [Lev75], height templates $T_i$ are given ( A1, B1, ... , A4, B4), and must be applied successively.

```
0 0 .      . 0 0      . 1 .      . 1 .
0 1 1      1 1 0      1 1 0      0 1 1
. 1 .      . 1 .      . 0 0      0 0 .
 A1         A2         A3         A4

0 0 0      1 . 0      . 1 1      0 . .
. 1 .      1 1 0      . 1 .      0 1 1
1 1 .      . . 0      0 0 0      0 . 1
 B1         B2         B3         B4
```

For one iteration, all the point of an image I coresponding to the template $T_i$ must be remove to perform the image J ( ¬ , |, & stand respectively for negation, logical or and logical and ) :

$$\begin{aligned} J &= I \,\&\, (\overline{T_i(I)}) \\ &= \overline{\overline{I} \mid (T_i(I))} \\ &= t1(t2(I)) \end{aligned}$$

So, this operation is the composition of two NCL $t1_i$ et $t2_i$, definied as :

$$\begin{aligned} t1_i &= \neg \\ t2_i &= \neg + T_i \end{aligned}$$

The application of the height template $T_i$ are implemented by NCL composition. So, the main loop of this pseudo-euclidian skeletonization can be perform by the smart sensor.

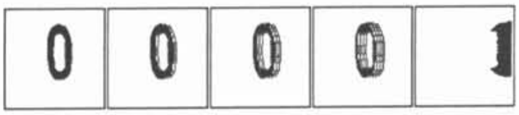
## An NCL counter.

Some global operations may be done in the smart sensor itself. In this paragraph, we will describe a realization of a counter with a composition of NCL operations. In the resulting image of counter algorithm, all the black pixels will be concentrated upon a border of the sensor. To count the number of black pixels, we only use the output of the number of black points along the bounds of the sensor.

The projection of the binary picture I upon the bound B of the sensor, translate all the black pixels with a given direction GD, up to the resulting image J, where all the black pixels are concentrated on B. This algoritm is presented in [Toffoli87] . Here is a NCL equivalence. For a projection from est to west, NCL p is as :

$$\begin{aligned} p &= 1\,0\,x + x\,1\,1 \\ &= \mu 1 + \mu 2 \end{aligned}$$

The two template $\mu 1$ and $\mu 2$ represent respectively a progression of one unit to the right, and the meeting with an obstacle. The projection consist of iteration of p, up to a constant image.



initial image, 3 first iterations and result image

**Fig.1 Projection p applied to an "O".**

All the Freeman vector projection may be given by rotation of p. These projections will use a reduced support ( 3 x 3 pixels ). The projection p' from north-west to south-est is :

$$p' = \begin{matrix} 1 & x & x \\ x & 0 & x \\ x & x & x \end{matrix} + \begin{matrix} x & x & x \\ x & 1 & x \\ x & x & 1 \end{matrix}$$

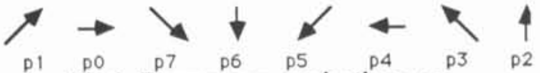


**Fig.2 Elementary projections $p_i$.**

If no border constraint exists, black propagated pixels will progressively diseapear (translation effect of $\mu 1$). Contrarily, if one border B is black, B will be an obstacle ( effect of $\mu 2$ ). Figure 3 is the illustration of the value of B. Black points are "1" pixels.

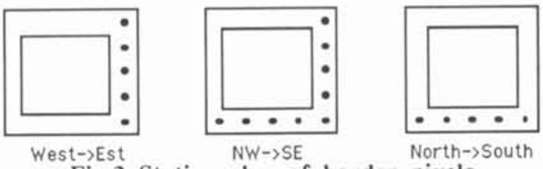


**Fig.3 Static value of border pixels.**

If n is the number of pixels of I, and $L = \sqrt{n}$ the width of the retina, the number of iterations is $\sqrt{n}$.

Now consider a black semi-plane of slope α. We illustrate in figure 4 the stability region $R_{p0}$ of projection $p_0$.

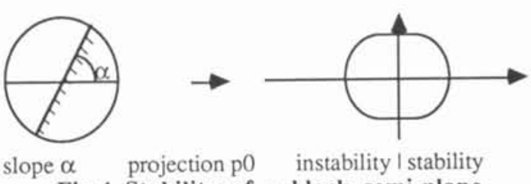


**Fig.4 Stability of a black semi-plane.**

If we consider a composition of projection pi, then the stability region of semi-planes will be the intersection of each region $R_{pi}$. The composition of elementary projections $p_i$ are used, to concentrate all the black pixels of the binary picture I upon a bound of the retina. The method consists in operating one or more cycles of n projections $c_k = (p_{1,k} \circ .. \circ p_{n,k})$, and to iterate $c_k$ until convergence. The choice of the projections is critical. Figure 5 illustrate an invariant picture under two projections. The stability region of projections p5 and p7 is non empty , and is as :

$$\alpha \in [-\frac{\pi}{4}, \frac{\pi}{4}]$$

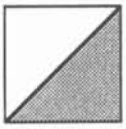


Fig 5. Invariant image under p7 and p5

The convergence will be obtained with projection without stability region, and a good convergence is experimentally got with c0 and c1 cycles. The counter algorithm needs four cycles. Figure 6 is the original halftoned picture. Figure 7b is the four resulted image (after cycles c0, c1, c0, c1).

$$\begin{aligned} c0 &= p0 \circ p7 \circ p6 \\ c1 &= p4 \circ p5 \circ p6 \end{aligned}$$

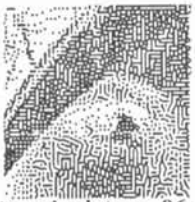


**Fig 6. Halftoned picture 96x96**

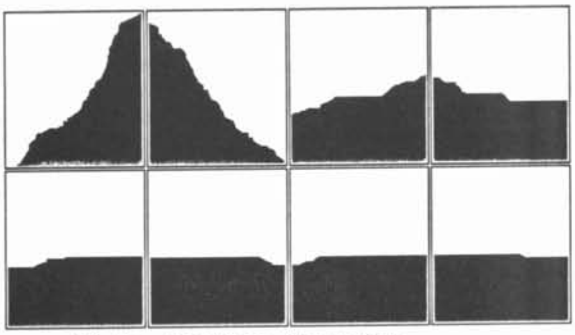
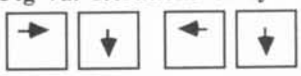


**Fig 7a. Iterations of cycles :**



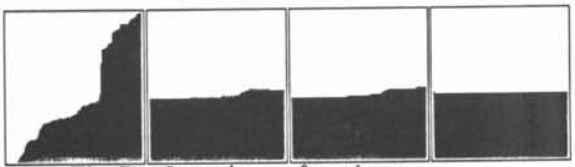
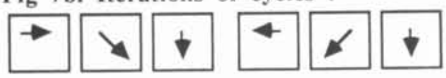


**Fig 7b. Iterations of cycles :**



## Conclusion.

We have described the architecture of a monolithic parallel boolean picture processor, used as a mesh array smart sensor, and some algorithms implemented on it. We have presented the implementation of a local operation, i.e. the main loop of the pseudo-euclidian skeletonization. We have proved that a global binary counter, as a control operation for pattern recognition, can be made with this processor. A 50 mm2 area ( 60x60 PE ) chip has been laid out in a 2 µ 2 metal CMOS technology .

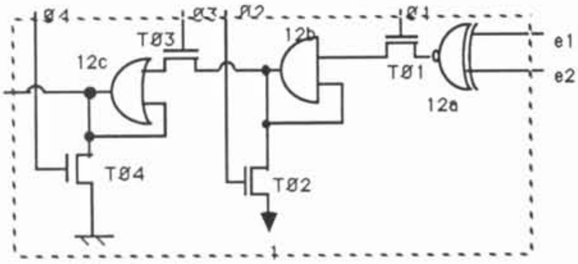


Fig. 8 : PE structure.

## Acknowledgements.

This work has been supported by grants from DRET and from SEFT. We thank B. Dugas and A. Lanusse for helpful discussions.

## References.

[Baker88] A flexible and intelligent system for fast measurements in binary images for in-line robotic control E.Baker, J.J.Gerbrands in Real-Time Object Measurement and Classification, Ed. A.K. Jain, Springer Verlag, NATO ASI Series F, Vol. 42, Berlin 1988, pp. 25-40

[Duff 86] M.J.B.Duff, T.J.Fountain. Cellular logic image processing. Academic press, London, 1986

[Garda 85] P.Garda, B.Zavidovique, F.Devos Integrated cellular array performing neighborhood combinatorial logic on binary pictures. 11th ESSCIRC, 16-18 sept. 1985, Toulouse [Duff 67] M.J.B.Duff, B.M.Jones, L.J.Townsend. Parallel processing pattern recognition system. Nucl. Instr. Methods, 52, 284-288, 1967

[Garda86] Rétine intégrée à réseau de processeurs F.Devos, P.Garda, B.Zavidovique. Brevet CNRS/ANVAR N°4 85 09256, 18 dec. 1986 (U.S. extension under way)

[Klette80] Parallel operations on binary images R. Klette CGIP, Vol. 14, pp. 145-158, 1980

[Person88] A pipelined image analysis system using custom integrated circuits E. Persoon IEEE PAMI-10, N°1, Jan. 1988, pp. 110-116

[Reddaway 79] S.F.Reddaway. The DAP approach. Infotech state of the Art Report on Supercomputers. Infotech Ltd, Maidenhead, 1979, Vol.2, pp. 309-329. VLSI System design. 1981

[Reeves 80] A.P.Reeves. On efficient global information extraction methods for parallel processors.CGIP, 14, 2, pp. 159-169, Oct. 1980

[Reeves84] Parallel computer architectures for image processing A.P.Reeves CVGIP, 25, pp. 68-88, 1984

[Toffoli87] Tommaso Toffoli, Norman Margolus. "Cellular Automa Machines" A new environment for modeling. M.I.T. Press, 1987.

[YOTE88] An automatic wafer inspection systems using pipelined image processing techniques H.Yoda, Y.Ohuchi, Y.Taniguchi, M.Ejiri IEEE PAMI-10, N°1, Jan. 1988, pp. 4-16

[LEV75] Arcelli,Cordella,Levialdi "Parallel thinning of binary pictures". Electronics letters, Avril 1975, Vol 11 No. 7.

[GRZ88] "Une rétine automate cellulaire programmable" P.Garda, A.Reichart, B.Zavidovique. To appear in "Traitement du Signal" 1988.