Real time phase-slopes calculations by correlations using FPGAs

J. Trujillo Sevilla[a], M.R. Valido[a], L.F. Rodríguez Ramos[b], E. Boemo[c], F. Rosa[a], J.M. Rodríguez Ramos*[a]

[a]Universidad de La Laguna, Facultad de Ciencias Físicas, 38202, Spain
[b]Instituto de Astrofísica de Canarias, 38200 La Laguna, Spain;
[c]Universidad Autónoma de Madrid, Spain

ABSTRACT

ELT laser guide star wavefront sensors are planned to handle an expected amount of data to be overwhelmingly large (1600x1600 pixels at 700 fps). According to the calculations involved, the solutions must consider to run on specialized hardware as Graphical Processing Units (GPUs) or Field Programmable Gate Arrays (FPGAs), among others.

In the case of a Shack-Hartmann wavefront sensor is finally selected, the wavefront slopes can be computed using centroid or correlation algorithms. Most of the developments are designed using centroid algorithms, but precision ought to be taken in account too, and then correlation algorithms are really competitive.

This paper presents an FPGA-based wavefront slope implementation, capable of handling the sensor output stream in a massively parallel approach, using a correlation algorithm previously tested and compared to the centroid algorithm. Time processing results are shown, and they demonstrate the ability of the FPGA integer arithmetic in the resolution of AO problems.

The selected architecture is based in today's commercially available FPGAs which have a very limited amount of internal memory. This limits the dimensions used in our implementation, but this also means that there is a lot of margin to move real-time algorithms from the conventional processors to the future FPGAs, obtaining benefits from its flexibility, speed and intrinsically parallel architecture.

Keywords: Adaptive optics, wavefront phase slopes, correlation algorithm, centroid algorithm, Sack-Hartmann sensor, FPGA, real-time processing.

1. INTRODUCTION

The Shack Hartmann sensor is widely used in adaptative optics to get an estimation of the wavefront at the telescope pupil, this is usually achieved by calculating the centroid shift on each subpupil, which is proportional to the wavefront phase slope. If reference sources are extended, not localized, the shift calculation by simple centroids loses effectiveness, and the correlation calculation is necessary.

The objective of this work is the development of a FPGA based system for the calculation of the centroid shifts making use of correlation. This represents an alternative to the calculation of simple centroids.

Therefore, if we want to correct the aberrations at real time, this is, before the aberration changes (10ms typically), the implementation of a fast enough algorithm ii order to get it is mandatory.

This implementation would be based either in microprocessors, in which the routine could be stored in memory and executed sequentially by reading its instructions, or in parallel processing devices, such as GPUs or high density programmable logic such as FPGAs in which a specific hardware for this algorithm could be implemented.

*jmramos@ull.es; phone +34 922 318249; fax +34 922 318228;

In this work the choice was to make the implementation of the system entirely over FPGAs. Taking this into account, all the advantages coming from the image hardware processing we can reduce the processing time to a lower level than the atmospheric characteristic time.

FPGAs are high density application specific hardware devices, which lets the creation of specific hardware for any algorithm in a short time, without the needing of hardware verification. The architecture of these devices are based on a net of logic cells and a net of interconnect cells, every connection is controlled by a SRAM cell. Therefore, determining the functionality of any cell or the route between two components is made by writing in the corresponding SRAM cell. Any process requiring more than one operation could be time multiplexed over the same circuit maximizing hardware efficiency.

2. SHACK-HARTMANN SHIFTS USING CORRELATION

The calculation of centroid shifts through correlation requires two fundamental algorithms: correlation algorithm and parabolic fitting algorithm.

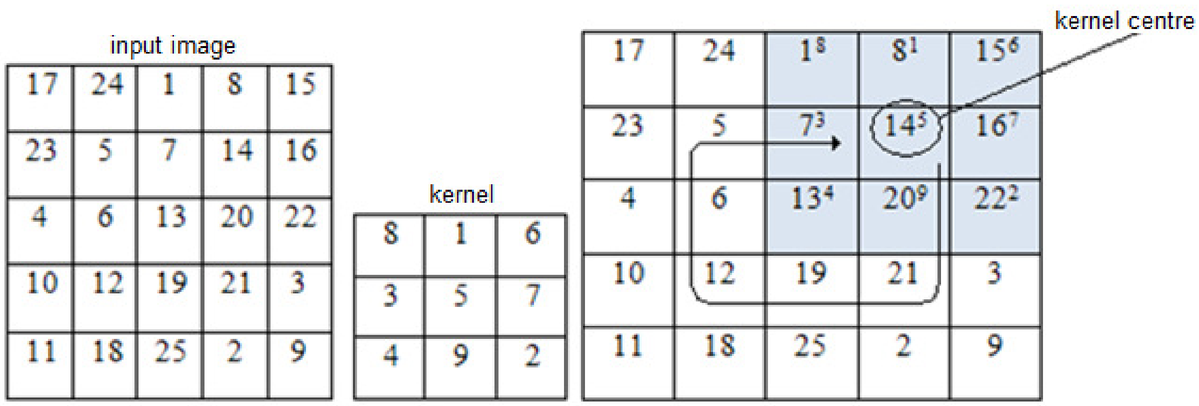
2.1 Correlation algorithm.

This algorithm is one of the best ways to estimate the general shift between two images and also is very effective in reducing the noise coming from remote pixels[1]. This shift will be the information used to estimate the analogous centroid shift on each image.

The bidimensional correlation, *C (x,y)* , needs two images as starting point, the input image (I) and the kernel image (K). The kernel image has to be smaller than the first one in order to make a correct calculation. The formula is expressed as follows:

$$C(x,y) = \sum_{i,j} I_{i,j}\, K(x_i + x, y_j + y) \quad (1)$$

It means that any output pixel represents a "weighted sum" of its neighbors pixels. The size of the output image will be defined by the sizes of the input images, figure 1. Output size will be size(I)-size(K)+1.



$$C(2,4) = 1\cdot 8 + 8\cdot 1 + 15\cdot 6 + 7\cdot 3 + 14\cdot 5 + 16\cdot 7 + 13\cdot 4 + 20\cdot 9 + 22 * 2 = 585$$

Fig. 1. Correlation calculation for pixel (2,4) at input image.

2.2 Correlation peak.

As mentioned, correlation output is another image, but to obtain the centroid shift we need five pixels. This is, the position of the output maximum value and its four adjacent pixels, as well as their values respectively. This output will be the input data for the next stage, parabolic fitting.

The correlation has the disadvantage of reducing the output size compared to the input image, this could result in a maximum value pixel placed in a border pixel, so one or two of the adjacent pixels wouldn't be calculated (figure 2). In this implementation, unknown pixels values are substituted by the peak value.

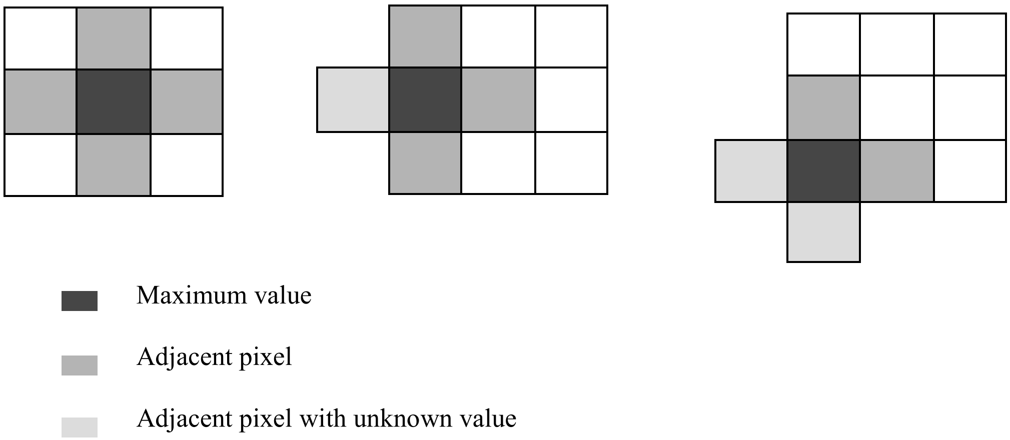


Fig. 2. Possible locations of correlation peak in a 3x3 output image. From left to right, centered peak, peak located on a border with one unknown pixel, pixel located on a border with two unknown pixels (worst case).

2.3 Parabolic fitting.

The objective of the parabolic fitting is to obtain the centroid shift with sub-pixel precision using as input data the correlation peak and its adjacents. The algorithm will fit a parabolic curve over each axis as seen on Fig. 3

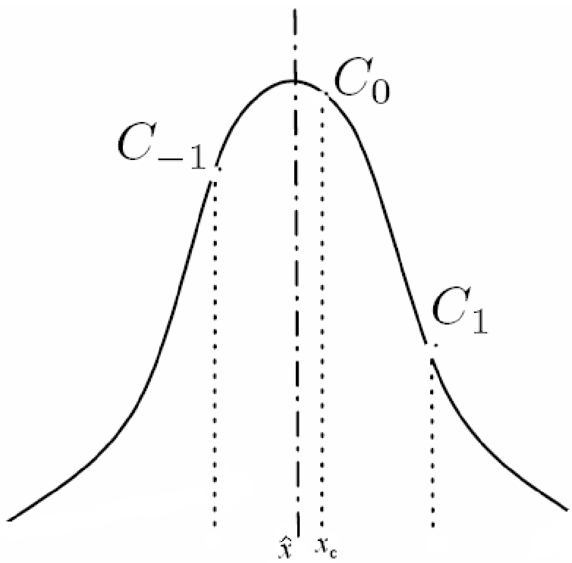


Fig 3: Representation of a parabolic fitting, where C_0 represents the peak value, C_{-1} represents the adjacent pixel on the left or below and C_1 represents the adjacent pixel on the right or above.

The parabolic fitting follows the formula:

$$\hat{x} = x_c - \frac{0.5[C_1 - C_{-1}]}{C_1 + C_{-1} - 2C_0} \qquad (2)$$

2.4 Shifts calculation process description.

The system will capture the images coming from the Shack-Hartmann wavefront sensor, the kernel image will be calculated online as every subpupil mean value, taking into account only the central part of it.

Next step consists in obtaining the reference for the final value of the centroid shift. Each centroid shift will be corrected with this reference. This reference value is obtained correlating a reference subpupil and the previously calculated kernel. This reference subpupil is initialized at system memory as a subpupil image without any kind of aberration.

Once the reference has been obtained, the centroid shift calculation starts. Each subpupil is then correlated with the kernel, and the obtained pixels are fitted to a parabolic curve, finally, for each centroid, the reference is subtracted.

3. IMPLEMENTED ARCHITECTURE. SYSTEM DESCRIPTION

The design was done entirely using VHDL and configured in a FPGA device. The system consists of several modules (Fig. 4), which will be detailed in the following paragraphs:

3.1 Acquisition.

This module captures the input images (Fig.5-A), storing only the parts of the image that contains the subpupils, storing each one as an individual image. This is accomplished by using a mask (Fig.5-B) previously initialized in memory. This mask will be 1, for the data containing the subpupils, and 0 for the data that won't be stored. This solution lets an improvement on BRAM utilization and a simplified data organization which makes the implementation of subsequent calculations easier (Fig.6).

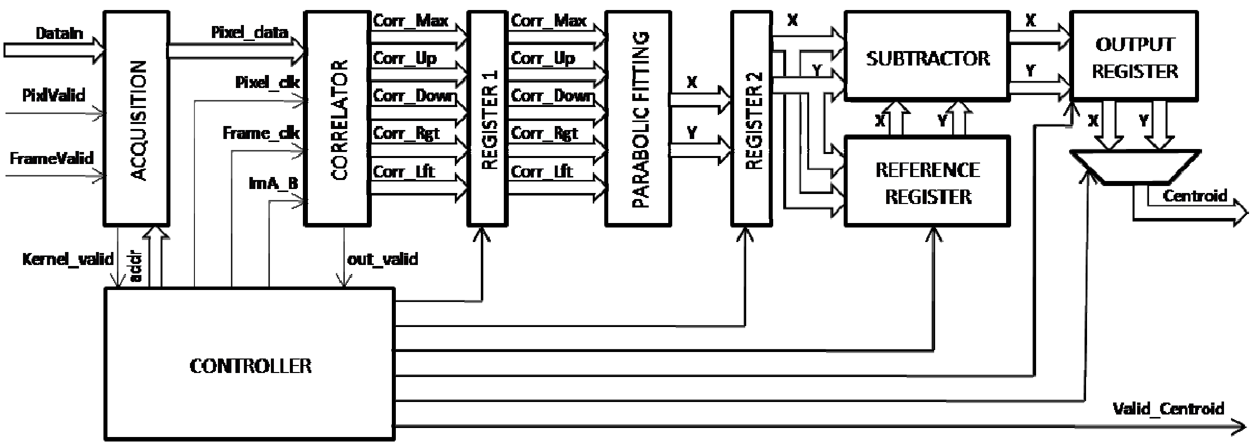


Fig 4. System architecture

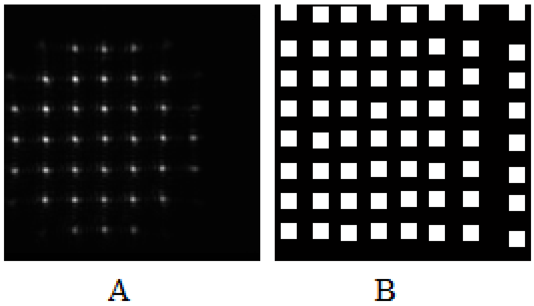


Fig 5. A: Shack-Hartmann sensor output image. B: Associated binary mask

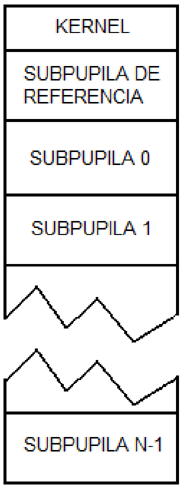


Fig 6. Input memory organization

This module also calculates the kernel, the calculation is done at the same time of the image acquisition by mean of an array of accumulators, one for each kernel pixel.

It also sends the image data to the correlation module; after the image captures any address in the valid range into its address port will result on the corresponding data.

3.2 Correlation.

This module computes the correlation between the Image I and the Kernel K. As first step, the module stores the kernel image. All the following image will be correlated with this one. Once the correlation is concluded, the module is ready to receive a new image.

3.3 Controller.

It consists of four Mealy's state machines that control several aspects of the design. It controls the reading address for the input memory on the acquisition module, the synchronism signals for the correlator module, the load signal for intermediate registers and the selection signal for output multiplexer.

3.4 Parabolic fitting.

This module was created taking into account that it will work during the time between two different correlations, therefore it had to be faster enough. So we decided the module should be totally combinational. This option has the disadvantage of having a combinational fixed point divider[2], which implies high resource utilization. The division algorithm schematic is shown in Fig. 7.

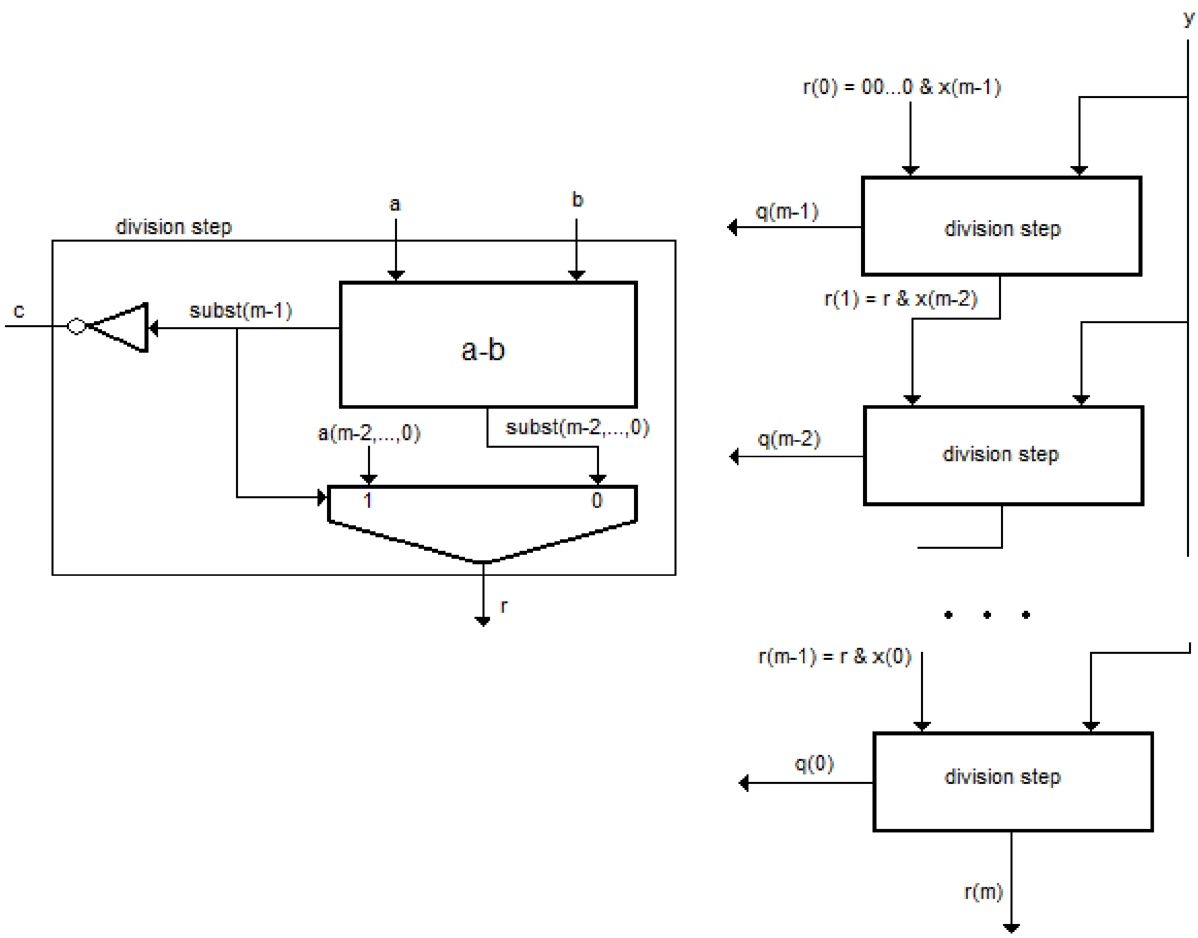


Fig 7. Fixed point combinational divider structure.

3.5 Other modules.

Subtractor module: It is a combinational fixed point circuit to subtract the reference to each result.

Output multiplexer module: Output data must follow the sequence X_1Y_1 X_2Y_2... X_NY_N, where N is the subpupil index. The data for both axis share the same port, so a multiplexer must select the data to present at the output. This module is managed from the controller module.

Intermediate registers: These registers were implemented by type D Flip-Flops without reset, activated by a rising edge at its load input.

- Register 1: Stores the output data from the correlator.
- Register 2: Stores the output data from the parabolic fitting module.
- Reference register: Stores the reference value.
- Output register: Stores the final values at the input of the output multiplexer.

3.6 Development environment.

The tool used to program, simulate and debug the design was Xilinx "Integrated Software Environment 8.2i" (ISE). As verifying tool Xilinx ChipScope Pro 8.2 was used. The hardware platform used was a Xilinx ML402 evaluation card, which includes a Virtex-4 SX35-10.

4. RESULTS

4.1 Processing time and resource utilization

In table 1 resource utilization and processing time to obtain all centroid shifts are detailed. 8x8 subpupils images were used with three sizes, 128x128, 256x256 and 512x512. Subpupil size grew proportionally. Note that for sizes greater than 256x256, processing time exceeds the 10ms of atmospheric characteristic time, therefore the closed loop correction will be ineffective.

	128x128 Sub:8x8 Ker:4x4	256x256 Sub:16x16 Ker:8x8	512x512 Sub:32x32 Ker:16x16	
Resources	**Used**			**Available**
Slices	4447	5431	9932	15360
Slice flip flop	1542	2096	5981	30720
LUTs	8250	10161	18607	30720
IOBs	29	29	29	448
FIFO16/RAMB16	7	22	73	192
GCLKs	16	16	16	32
DSP48s	4	5	6	192
PMCDs	1	1	1	4
Time(100 MHz)	664.2 µs	2.89 ms	18.1 ms	

Table 1. Resource utilization and processing time .

4.2 Wavefront recovery

In Fig.8, a comparison between wavefront recoveries is shown. The same wavefront map is obtained from a centroid algorithm and from our correlation algorithm implementation. The resemblance between the recoveries and the sampled original wavefront is quite high but with only one wavefront it is impossible to determine which one of the methods offers a better approximation. Only a statistic analysis could give enough data to obtain a conclusion about that.

Fig 8. From left to right, original wavefront, wavefront recovered by simple centroids, wavefront recovered by correlations.

5. FUTURE WORK

In order to reduce the processing time and then to undertake higher samplings, the most direct idea consists in adding several FPGAs running in parallel and then much more correlators could be implemented in. Nevertheless, using exactly the actual implementation over the modern Xilinx virtex-5 greater samplings can be undertaken inside the atmospheric characteristic time.

Next step to improve the wavefront phase recovery consists in modify the correlation kernel. It should be obtained from the average of the precedent subpupil-frames.

Finally, this algorithm will be easily adapted to the CAFADIS camera, providing the wavefront phase associated to extended objects. This will be the starting point for tomographical phase reconstruction using the CAFADIS camera (Rodríguez-Ramos *et al*.[3]).

6. ACKNOWLEDGMENTS

This work has been partially supported by “Programa Nacional de Diseño y Producción Industrial" (Project DPI 2006-07906) of the “Ministerio de Educación y Ciencia" of the Spanish Government, and by “European Regional Development Fund" (ERDF).

REFERENCES

[1] S. Thomas, T. Fusco, A. Tokovinin, M. Nicolle, V. Michau y G. Rousset. *Comparison of centroid computation algorithms in a Shack-Hartmann sensor.* Monthly Notices of the Royal Astronomical Society (2006).

[2] J. P. Deschamps, G. J. A. Bioul, G. D. Sutter. *Synthesis of arithmetic circuits.* Ed. Wiley-Interscience (2006).

[3] Rodríguez-Ramos J.M. , Femenía Castellá B., Fumero S. and Pérez Nava F. “*Wavefront and distance measurement using the CAFADIS camera*. SPIE Astronomical telescopes and Instrumentation Conference, Marseille, 2008.