World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:2, No:11, 2008

A Parallel Architecture for the Real Time Correction of Stereoscopic Images

Zohir Irki, and Michel Devy

***Abstract*—**In this paper, we will present an architecture for the implementation of a real time stereoscopic images correction's approach. This architecture is parallel and makes use of several memory blocs in which are memorized pre calculated data relating to the cameras used for the acquisition of images. The use of reduced images proves to be essential in the proposed approach; the suggested architecture must so be able to carry out the real time reduction of original images.

***Keywords*—**Image reduction, Real-time correction, Parallel architecture, Parallel treatment.

I. Introduction

THE stereoscopic images are images perceived of the same scene but acquired from different points of view. Often, because of a bad camera's positioning and/or of some technical defects related to their design, the acquired images need to be improved before sending them to later treatment stages.

Among these improvements, the correction which aims at replacing the two original images by two others, giving same photometric information on the perceived scene, but having properties adapted for the principal stages of stereovision: the 3D reconstruction and the mapping between correspondents.

The correction of the images can be done using pre-calculated tables of correction. The nature and the number of these tables are closely related to the temporary aspect of the used approach (off-line or real-time).

The off-line correction approach [1] consists of using, for each image, only one pre-calculated table in which for each corrected pixel is associated the distorted pixel which represents the approximation of the equation's system giving the relationship between a distorted and a corrected image.

In the opposite, the real time correction approach, which is an improvement of the off-line approach, makes use of two pre-calculated tables. The first gives information on the number of correspondents of each distorted pixel in the corrected image. The second table gives information on the emplacement of these correspondents in the corrected image [2,3].

Zohir Irki is with the Numerical Systems Laboratory of the Military Polytechnic School, BP 17 BEB, Algiers, Algeria (e-mail: zohir_irki@yahoo.fr).

Michel Devy is with le Laboratoire d'Analyse et d'Architecture des Systèmes/ CNRS, 07 Avenue du Colonel Roch, 31077, Toulouse, France and l'Université de Toulouse ; UPS, INSA, INP, ISAE ; LAAS-CNRS : F-31077 Tolouse, France (e-mail: :michel@laas.fr).

The number of the correspondents of a distorted pixel can be 0, 1 or higher, whereas the access to a physical memory block can take place only once in each cycle clock (in reading or writing mode). This fact makes difficult to carry out the real time aspect of the correction. The reduction of the original images is a solution to this problem [4]. In fact, the real time reduction of images makes possible the prohibition of data transfer but neither their acquisition or treatment [5,6]. How to reduce an image is a very important to guarantee the uniformity of the treatment and synchronisation between the different blocks ensuring the real time correction task.

In this work, we describe an architecture for the implementation of the real time correction approach proposed in [5, 6]. This implementation is to be done on an FPGA STRATIX 1S40F780C5. The proposed architecture must respect the technological limitation of the used FPGA, in particular the limitation of memory resources (3Mbits) and the frequency of frame grabbing (1 pixel each 25 ns).

The purpose of this work is to describe our proposed architecture, we avoid making comparisons between the off-line and the real time correction approaches. Results of the application of the two approaches are practically the same because the real time approach is derived from the off-line one [6].

This paper will be started by a brief description of the suggested real time correction approach. Thereafter, a description of the used method in order to carry out the real time image reduction is given. We will finish by describing the suggested architecture.

II. Principle of the Stereoscopic Images Correction

The correction of the stereoscopic images is a task which can be subdivided in two tasks [7]:

- The alignment of the images: this task consists of changing the original images so that a point perceived in the two images will be in the same line. This can be done easily by manipulating matrices representing the stereoscopic bench system.
- The correction of the radial distortions in the images: this is more complex than the first task. The problem of the real time correction of stereoscopic images is summarized to the real time correction of the radial distortions in the original images.

The radial distortions come from the defects of curve on the lenses of the camera's objective. These distortions appear in the image plan like a translation along the ray joining the projection of the optical center (not far from the center of the

image) and the considered pixel.

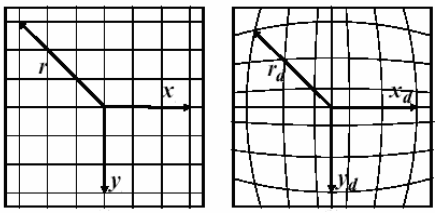

Fig. 1 Radial distortions

On the image plan, in metric coordinates, we distinguish:

- The ideal coordinates (x, y), which correspond to the perfect projection;
- The Distorted coordinates (x_d, y_d).

The relationship between these coordinates is given by the system of equations:

$$\begin{cases} x_d = x.\left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right) \\ y_d = y.\left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right) \end{cases} \quad (1)$$

Where:

- k_i : Are the coefficients of distortion on the metric coordinates. These coefficients are given following the calibration of the stereoscopic bench.
- $r^2 = x^2 + y^2$: Is the distance to the principal point.

The off-line correction [7] consists of calculating, for each pixel (u,v) of the corrected image, the position of the corresponding distorted pixel (u_d,v_d) of the original image. Then to allot the value of brightness's function (grey scale) of the distorted pixel to the corrected one. This operation consists of the sweeping of the corrected image and to solve the system of equations (1) for each pixel in order to determine the distorted position corresponding to the corrected position. In order to calculate the grey scale in the distorted position, we can proceed either by approximation [8], or by interpolation [4].

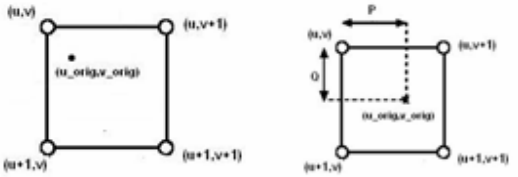

Fig. 2 Approximation and Interpolation

The memorization of the couples *(corrected pixel, distorted pixel)* based on the resolution of the system (1) will make it possible to build pre-calculated correction tables. These tables are called indirect because they make it possible to build corrected images starting from a sweeping of virtual initial corrected images. This sweeping is in order to find the correspondents in the distorted images before allotting the grey scale of these correspondents to the corrected pixels. This method is qualified as off-line because it requires to memorize a certain number of lines of each distorted image before beginning the treatment [2,3].

The real time approach is an improvement of the off-line approach [5, 6]. In this approach, we propose to build direct pre-calculated tables starting from indirect tables. The direct tables have two different natures.

The first table is intended to memorize, for each distorted pixel, the number of correspondents in the corrected image as well as the address, in a memory, where these correspondents are memorized if they exist. This first table makes it possible to classify the distorted pixel in two categories: **active** and **passive** pixels. An active pixel has at least one correspondent in the corrected image while a passive pixel has not any correspondent. The second table is intended to memorize, for each pixel active, the correspondents in the corrected image [2, 3].

The improvement of the off-line approach in order to adapt it for the real time treatment is done by two sweeping of the indirect tables.

The first sweeping makes it possible to build the first direct table called table of addresses. Initially, all the distorted pixels are supposed to be passive. During the first sweeping of the indirect table, to each corrected pixel correspond the distorted pixel which represents the integer approximation of the solution of the system of equations (1). This distorted pixel will thus be considered as active. During this sweeping also, we count how much time each distorted pixel is met. This number represents the number of correspondents of the active pixel. At the end of the first sweeping, the table of addresses will be built and its size is the number of the pixel of the distorted image.

The second sweeping allow us to determine, for each active pixel, the emplacement of its correspondents in a physical memory. This sweeping makes it possible to build the second direct table which size is the sum of the numbers of the correspondents of all the active pixels.

The manner with which the second pre-calculated table is built allows a serial treatment and this is not adapted for a real time treatment because of the unicity of the access to a physical memory in each cycle clock. Some improvements were made in order to have a parallel treatment allowing real time correction of stereoscopic images [5,6].

The first improvement consists of the building of the pre-calculated tables starting from reduced images. This makes it possible to write data in the physical memory dedicated to memorize the corrected image because a distorted pixel can have several correspondents. Image reduction will allow several accesses to the memory because that the treatment cycle clock will be higher than the system cycle clock. More details are in the next paragraph.

The second improvement consists of the limitation of the numbers of correspondents so that the quality of the corrected image will not be appreciably affected. In our work, the experiment showed that the fact to limit the number of correspondents to 3, has not remarkable influence on the

corrected image and practically on influence on the disparity image when considering stereoscopic images [5,6].

When the number of correspondents is limited to 3, the correspondents will be memorized on 3 memory boxes having the same access address but belonging to 3 physical memories. Under these considerations, the first pre-calculated table will be intended to memorize, for each active pixel, the address where correspondents are stored. The first pre-calculated will always allow classifying distorted pixels into passive and active and this by using only one bit (0 for passive and 1 for active).

For more details on the passage from the off-line to the real time approach, we refer [2], whereas [5,6] give additional details on the improvements made to order to adapt the approach to the real time treatment.

III. Images Reduction

The fact that an active pixel has 3 correspondents in the corrected image makes that the memorization of the corrected image (writing on the physical memory dedicated to memorize the corrected image) requires at least 3 system clock cycles. In this case, data transfer represents a problem. This task is carried out in each cycle clock (video signal acquisition). So in each cycle, we must treat new 3 data and this will make the treatment impossible. The solution consists of the real time acquired image reduction (in this case, the used pre-calculated tables must be built starting from reduced images).

Several methods can be used to reduce an image. But in all these methods, a pixel replaces a window made up of (f_l, f_c) pixels. Where f_l represents the factor of line reduction and f_c represents the factor of column reduction.

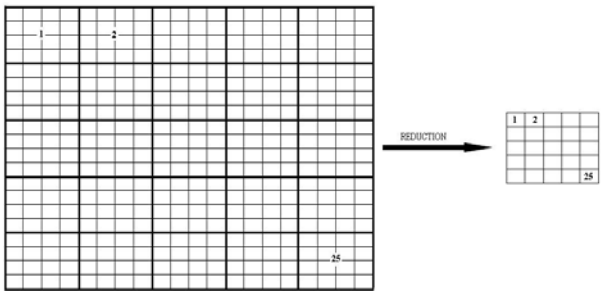

Fig. 3 Image Reduction

The first method which can be used consists of replacing the original image whose size is $[N_l * N_c]$ by a reduced image having a size of $\left[\frac{N_l}{f_l} * \quad \frac{N_c}{f_c}\right]$. The grey scale in the position (u_r, v_r) will be the mean of the grey scales of the pixels belonging to the window. This method requires subdividing the original image into windows, the size of each windows is (f_l, f_c), and then to calculate the mean of grey scales of each window before allotting the result to the pixel having the same order in the reduced image as the order of the window in the original image. We need thus to memorize data. So this method is not adapted to a real time treatment.

A second method consists of allotting the grey scale of the first pixel of each window to the pixel which replaces it in the reduced image (reduction by regular sampling, Fig. 4) [1]. This method does not require memorize data because once the grey scale of the first pixel is acquired, it will be immediately allotted to the corresponding pixel. Nevertheless, this method represents a time management problem. In this method, it will be considered only one line among f_l lines and one column among f_c columns.

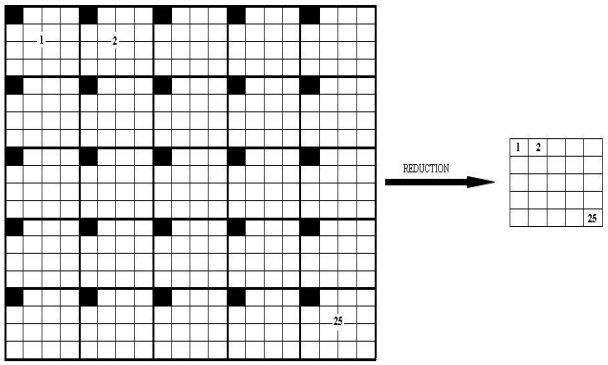

Fig. 4 Reduction by regular sampling

Let consider the example of a camera which provides an image which size is [100*100]. We want to reduce the acquired image with $f_l = f_c = 4$. The acquisition frequency is a pixel each 25 Ns.

The grey scale of the first pixel of the reduced image is obtained immediately after the acquisition of the first pixel of the original image. The other values on the first line are obtained with a frequency equal to 1 pixel each 100 Ns. Once the first line acquired, we will have to wait 7.5 µs before obtaining the first grey scale of the first pixel of the second line of the reduced image. We notice that in this method, a problem of uniformity and time management are confronted.

In order to standardize the treatment, we propose a third method which is reduction by non regular sampling [4]. In this method, we define a total reduction factor:

$$F = f_l * f_c \tag{2}$$

When we consider an image, a pixel is located by its position line and its position column (i, j). In the suggested approach, a pixel will be located by its position P in a vector:

$$P = (i-1) * N_c + (j-1) \tag{3}$$

Like this, the original image will be represented by a vector in which the first position will be occupied by the first pixel. This position will have the range 0. The last position having the range $N_l * N_c - 1$ will be occupied by the last pixel. The reduced image will be also memorized in a vector composed

of $\frac{N_l * N_c}{F}$ components.

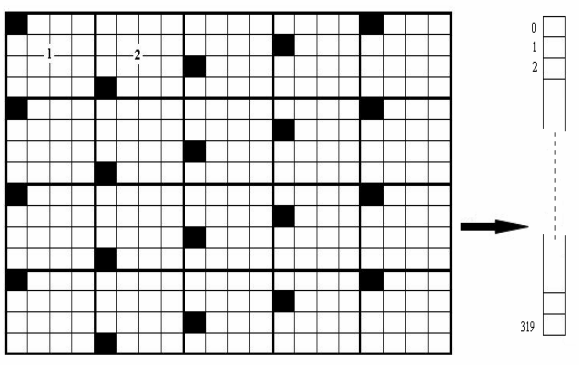

Fig. 5 Reduction by non-regular sampling

In this approach, each window will be replaced by a pixel belonging to it. But these pixels will not be taken uniformly. Only pixels which their positions check the following constraint will be retained:

$$P = F * a + F - 1 \quad (4)$$

A problem can appear when two pixels belonging to the same window check the equation (4). For that, we must impose the number of column of the original image so that only one pixel from each window could check the constraint. The fact of using a CMOS camera [9] allow us to avoid this problem. The number of the columns N_c of the original image must be such as only $f_l * N_c$ is multiple of F .

For a reduction factor $F = 16, (f_l = 4, f_c = 4)$, the pixels to be retained are those which check the constraint $P = 16 * a + 15$. The number of columns of the original image must be such as only $4 * N_c$ is multiple of 16.

This approach will make it possible to standardize time management. Data acquisition will still at the frequency of 1 pixel each 25 Ns, but the transmission of the acquired data from the camera to the interface treatment should be carried out at the frequency of 1 pixel each 400 Ns.

In this approach, it is very important to know the order of memorizing of the pixels of the reduced image.

Let us take as example the image of Fig. 5, the first four line of the original image will contribute to obtain the first line of the reduced image composed of 5 pixels. The order of memorizing of these pixels is as follows: (1,5,4,3,2).

This order, which is a function of the size of the original image as of the reduction factor, will be maintained until the reduction of the entire original image.

IV. The Proposed Parallel Architecture

The purpose of the proposed parallel architecture is to implement the suggested real time correction approach on an FPGA component type *STRATIX 1S40F780C5*. This component belongs to the altera family, therefore the ***Quartus*** environment is used to develop the architecture.

The proposed architecture must ensure the realization of the following tasks:

1. The real time reduction of the original images (acquired starting from a CMOS camera);
2. The classification of the distorted images pixels, then the determination of the addresses where correspondents of actives pixels are memorized;
3. The determination of the exact positions of correspondents in the corrected images;
4. The assignment of the grey scale of the active distorted pixel to all its correspondents, and then the memorization in the physical memory intended to memorize the corrected images.

The real time reduction of images allows us to parallel the treatment which is of primary importance at stages 3 and 4. In opposite, since only one physical memory is needed to memorize the corrected image, stage 5 must be done by a serial way. Therefore in this stage, we must turn over towards the serial treatment.

The proposed architecture makes use of modular programming. Each stage is carried out by a module. The programming was made using the VHDL language.

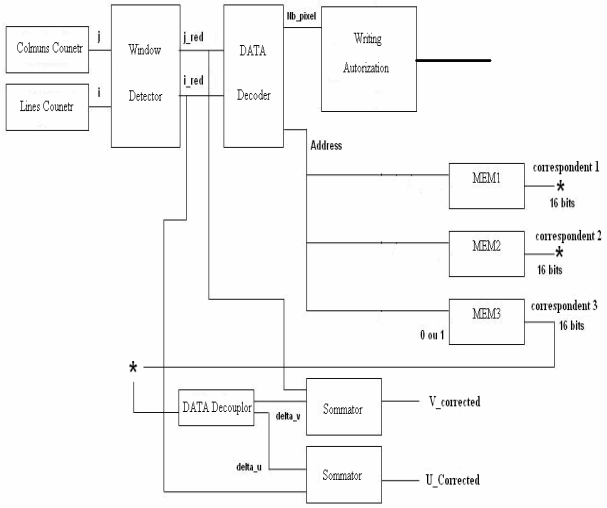

Fig. 6 Synoptic scheme of the proposed architecture

A. Image Reduction

The "reduction" term does not mean the construction of a reduced image but it is a term with which we mean the determination of the position of a pixel belonging to a non reduced image in a reduced one.

Image reduction will be done by taking a pixel among 16 pixels, this mean that it will be ensured by the detection of the window to which the sampled pixel belongs. The reduction module called "window detector" is based on simple counters and division blocks. A counter of pixels makes it possible to determine the order of the image sweeping. To insure a good sampling, we must generate an impulse once each 16 clock

cycles. This impulse will be used as clock to the pixels counter and thereafter allows the control of all the architecture.

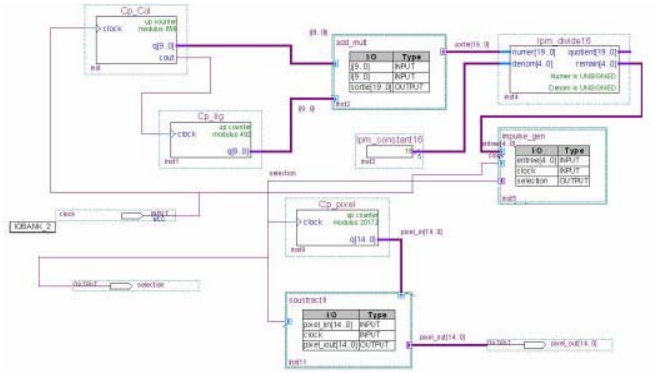

Fig. 7 Sampling clock generation

The window detector has also an output composed of 15 bits being used to point a memory box in which are memorized the correspondents of the considered pixel. This output is the output of the pixels counter but for initialization reasons, we must extract 1 from it.

The position of the pixel in the reduced image will be given by the coordinates (u_r, v_r) which are the outputs of the dividers by 4 having (u, v) as inputs. The dividers by 4 are controlled by the principal clock of the system and not by the sampling clock. So, the current value of (u_{red}, v_{red}) must be maintained for the other steps of treatment (calculation of correspondents). For that, we must add a module intended to realize the blocking of these values.

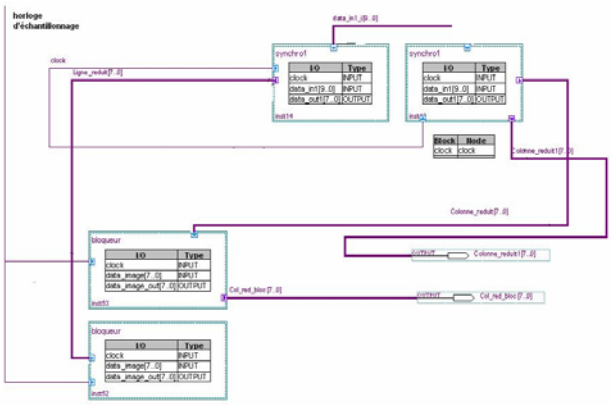

Fig. 8 Blocking of the reduced values

The used blockers must be controlled by the sampling clock. This makes that the value of (u_{red}, v_{red}) will change only once each 16 cycles of the principal clock.

B. Pixels Classification

This task consists of recovering data relating to each pixel then to extract from it the number of its correspondents in the corrected image. For memorizing these data, we need an address bus composed from15 bits. The QUARTUS library provides preset memories which capacities are power of 2. The addresses table is composed from 20172 words of 16 bits, so we used two memories in order to memorize all these words. The first memory of 16384 words (16384=2^14) thus its address bus is composed from 14 bits. The second of memory of 4096 words, therefore addressable via an address bus of 12 bits. The association of these two memories will give a memory addressable via an address bus of 15 bits. The 15th bit will be used for the selection of one among the two used memories and will make it possible to know from where data has to be recovered.

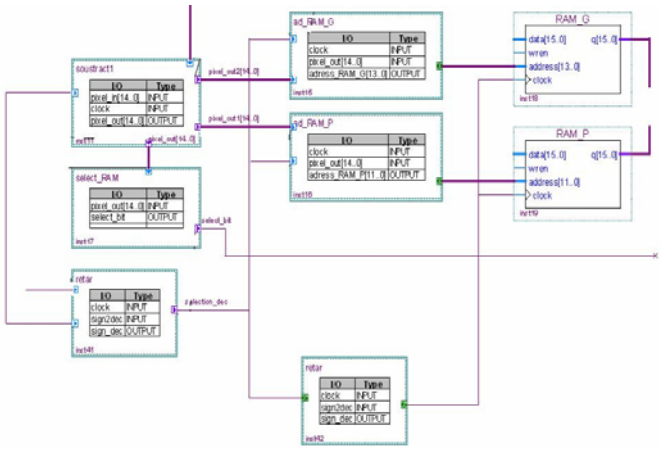

Fig. 9 Addresses table management

For synchronization reasons, we introduce a module intended to shift the sampling clock. This module will give to data a sufficient propagation time.

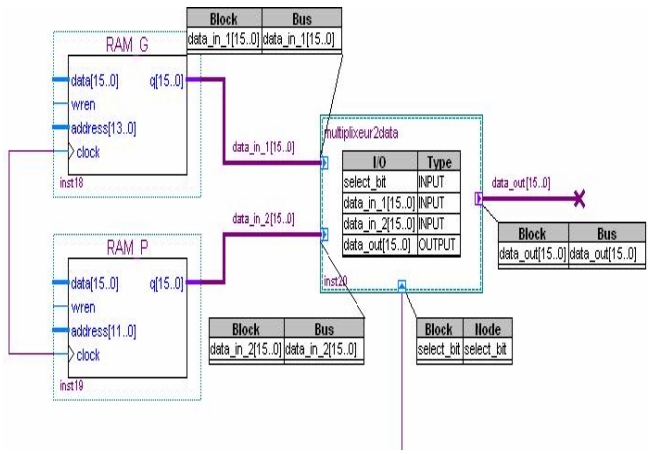

Fig. 10 DATA recuperation

C. Determination of the Positions of Correspondents

After the determination, for a distorted pixel, of the address where its correspondents are memorized, it is necessary to jump to another step of treatment. This step consists of the calculation of the position of these correspondents in the corrected image and to thereafter allot the grey scale of the

original pixel to all its correspondents in the corrected image.

As the number of correspondents is limited to three, three memories should be envisaged.

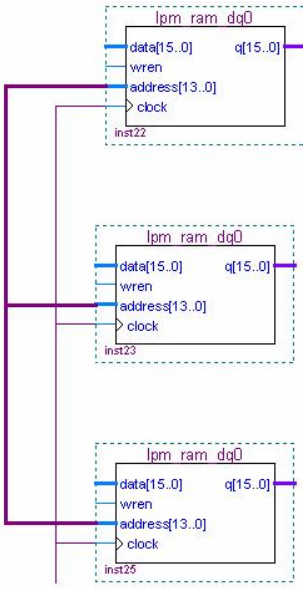

Fig. 11 Correspondents table organisation

Since that all the pixels do not have three distinct correspondents, and then it is possible that three memory boxes, having the same address, contain the same data. In this case, data is a duplication of the last correspondent. We can speak so about a simple, double or triple correspondent.

A correspondent is characterized by two displacements $(\Delta u, \Delta v)$. From these displacements, it is necessary to calculate the position (U,V) in the corrected image. For that, starting from memorized data, Δu , Δv and the sign of the displacement are determined. A simple addition operation allows to calculate precisely the positions U and V .

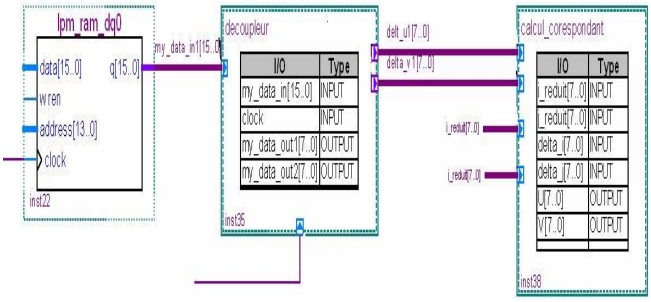

Fig. 12 Correspondents calculus

The module called "decoupleur" is charged to determine displacements Δu and Δv . These displacements are coded using 8 bits. When the 8th is null, displacement is positive while displacement is negative when this bit is set.

The "correspondents calculus" module has two 8 bits outputs (U and V) because the output image (corrected image) is a reduced image whose dimensions can be coded using 8 bits (lower than 255). Starting from these outputs, we must determine where the corrected pixel is memorized.

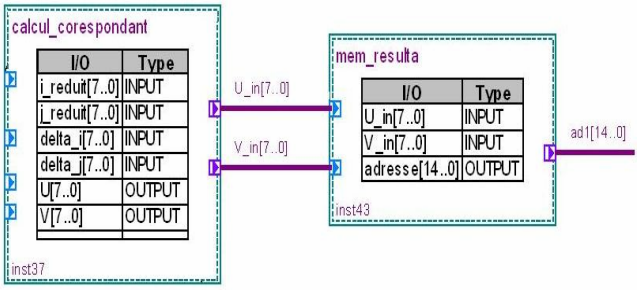

Fig. 13 Memorization address calculus

D. Memorization of the Corrected Image

The memory intended to memorize the corrected image is a unique memory on which we must write three times to each clock treatment cycle. This writing must be done on a serial way in opposite to the treatment which is done on a parallel way. It is necessary thus to generate a clock which controls the memorization of the corrected image so that each data must be written in the right emplacement. For that addresses must be transferred one after one.

To ensure this task, a data fusion is made in order to build a unique 45 bits data starting from three 15 bits addresses. At each writing clock cycle that, we must generate, we must rotate the combined 45 bits data. Like this, at the first writing clock cycle, we write in the first memory box. After the rotation, the second address becomes the first, the third becomes the second and the first becomes the third. At the next writing clock cycle, we write in the second memory box and the process will be repeated once in order to write in the third memory box.

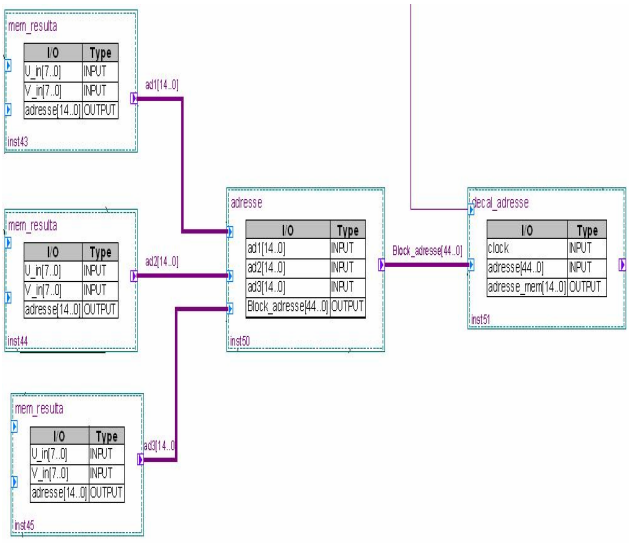

Fig. 14 Shift management for memorization

In this way, we will be able to correctly manage the process of the corrected image memorization. For a better synchronization of the various steps of treatment, we introduce delay modules which shift the treatment clock. These delay modules make that a task will not begin before the end of the task which precede it.

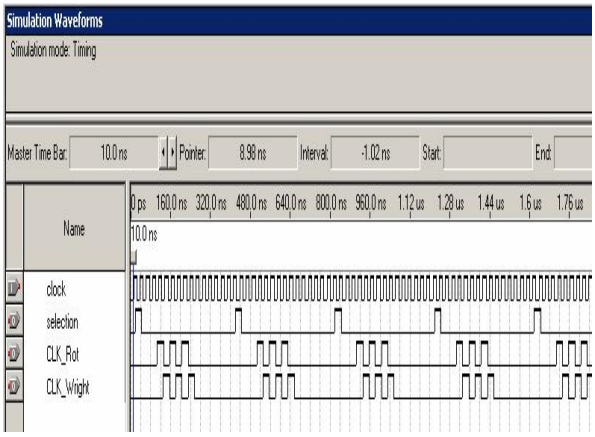

Fig. 15 Temporal analysis

The temporal analysis (Fig. 15) illustrates that the treatment process takes 9 cycles of the principal clock. So the real time aspect is strictly respected because the treatment cycle is equal to 16 times the principal cycle.

V. Conclusion

In this paper, we described an architecture which will allow the implementation of the real time image correction. This correction is based on the use of pre-calculated tables. The fact of considering the alignment of the stereoscopic images when constructing these pre-calculated tables adapts the method to the real time correction of stereoscopic images. The duplication of the proposed architecture makes it possible to implement the method on an FPGA component.

In the suggested architecture, the distorted original image, whose size is large, will be reduced for then using the reduced image in treatment operations. The reduction of the original image seems to be essential for the temporal management of treatments.

The suggested architecture is based on the use of pre-calculated tables which associate to each input several parameters necessary to carry out the desired task.

This architecture makes use of the parallel treatment. Indeed, it is this type of treatment which allows the real-time aspect. A serial treatment can be envisaged, but in this case, we will not be able to guarantee the real-time aspect.

Further works consists of the real implementation of the proposed architecture on an FPGA component and the evaluation of the really required resources.

References

[1] A. Naoulou: "FPGA Based Architecture for Real Time Computation of the Census Transform and Correlation in Various Stereovision Contexts'', LAAS report December 2004.
[2] Z.Irki, M. Devy, K. Achour : "Une Approche pour la Rectification Temps Réel des Images Stéréoscopiques'' ; CGE'05, EMP 2007.
[3] Z. Irki, M.Devy, P. Fillatreaud, J. L. Boizard, "An Approach for the Real Time Correction of Stereoscopic Images", ECMS & Doctoral School, (EDSYS, GEET), May 21-23, 2007, Liberec, Czech Republic.
[4] Z.Irki: Rectification Temps Réel des Images Stéréoscopique'', Rapport de Stage, LAAS 2006.
[5] Z.Irki, M.Devy, K.Achour, "The Real Time Correction of Stereoscopic Images: From the Serial to a Parallel Treatment". Accepted in ICTTA 08; Syria.
[6] Z.Irki, M.Devy, K.Achour, "La Rectification Temps Réel des Images Stéréoscopiques : Du Traitement Série à un Traitement Parallèle", CVA07, Novembre 2007 Tizi Ouzou, Algérie.
[7] M.Devy: "Présentation de la stéréovision passive''; Rapport interne LAAS 2005.
[8] V.Lemonde : ''Stéréovision Embarquée sur Véhicule: de l'Auto-Calibrage à la Détection d'Obstacles'' ; Thèse de Doctorat, INSA Toulouse, Novembre 2005.
[9] Digital monochrome quad speed CMOS progressive scan camera; operation manuel version 1.0.

Zohir Irki was born in Médéa (Algeria) in 1976. In 1994, He obtained his baccalaureate and in 2001 he finished his engineering studies (Electrical engineering, Automatic) from the Military Polytechnic School of Algeria. In January 2005, he had his magister Diploma (Control and command) always from the military polytechnic school. Actually, he is preparing a thesis on the Real-time stereoscopy for obstacle detection in collaboration with the LAAS/CNRS, France.

Michel Devy got his degree in Computer Science Engineering in 1976 from IMAG, in Grenoble (France). He received his Ph.D. in 1980 from LAAS-CNRS in Toulouse (France). Since 1980, he has participated to the Robotics and Artificial Intelligence group of LAAS-CNRS; his research is devoted to the application of computer vision in Automation and Robotics. He has been involved in numerous national and international projects, about Manufacturing Applications, Mobile Robots for space exploration or for civil safety, 3D Vision for Intelligent Vehicles or Medical applications, ... He is now Research Director at CNRS, head for the Perception Area in the Robotics Group of LAAS-CNRS and his main scientific topics concerns Perception for Mobile Robots in natural or indoor environments. He is author or coauthor of about 100 scientific communications.