High speed camera with embedded image processing

Michel Paindavoine, Julien Dubois, Romuald Mosqueron,
Barthelemy Heyrman, Jerome Dubois, Dominique Ginhac

Universite de Bourgogne, Laboratoire Le2i - UMR CNRS 5158
Aile des Sciences de l'Ingenieur, BP 47870 - 21078 Dijon cedex, France
email:*paindav@u-bourgogne.fr*

ABSTRACT

High-speed video cameras are powerful tools for investigating for instance the dynamics of fluids, human or biological movement analysis. In the past years, the use of CMOS sensors instead of CCDs have made possible the development of high-speed video cameras offering digital outputs, readout flexibility and lower manufacturing costs. In this field, we designed a new fast CMOS camera with a1280x1024 pixels resolution at 500 fps. In order to transmit from the camera only useful information from the fast images, we studied some specific algorithms like edge detection, wavelet analysis, image compression and object tracking. These image processing algorithms have been implemented into a FPGA embedded inside the camera. This FPGA technology allows us to process fast images in real time. In order to improve performances of our high speed camera, we designed also two new devices: a concept of fast smart camera and a high speed CMOS image sensor with embedded analog image processing.

Key words: CMOS Image Sensor, FPGA, Embedded Image Processing, High-speed Video

1. INTRODUCTION

For the fifteen last years, our laboratory has worked in high-speed video systems areas[1,3] and obtained results for biological applications like real time cellular contractions analysis[2] and human movement analysis.[4] All these developments were made using CCD imaging technology from Fairchild and FPGA technology from Xilinx.[5] The main goal of our systems was to provide, with a low price, high-speed cameras (500 frames per second) using standards CCD devices in bining mode, with a pre-processing FPGA module connected to a PC-compatible computer.

In the past years, the use of CMOS sensors instead of CCDs have made possible the development of industrial high-speed video cameras offering digital outputs, readout flexibility and lower manufacturing costs[6–11] Two main limitations of these systems can be discussed. First, the huge data flow provided by the sensor cannot be easily transferred or processes and has generally to be stored temporary in a local fast RAM memory. This RAM is size limited so the recording time in the camera is only a few seconds long. Using an image compression approach, we developped an alternative solution that allows continuous recording.[12]

Second, in order to execute in real time tracking or object recognition, it is necessary to process information from the image data flow and for this embedded real time calculations have to be implemented inside the camera. Our aim, here, is to describe the implementation of real time image processing inside our new high-speed video system.

This paper is organized as follow. Our high-speed camera is described in Section 2. The studied image processing algorithms applied to a biomechanics application are introduced in Section 3. Then, we present the implementation of these algorithms inside the FPGA in Section 4. In order to improve performances of our high speed camera, we conducted two developments: a new concept of fast smart camera and a new high speed CMOS image sensor with embedded analog image processing. These developments are described in section 5. Finally conclusions and perspectives are drawn in Section 6.

2. HIGH-SPEED CAMERA DESCRIPTION

Our high-speed camera is based on a high-speed CMOS image sensor from Micron[13] for image acquisition and on a FPGA device from Xilinx[5] for image processing. In order to transmit fast images to a PC computer, an image compression algorithm has been implemented inside the FPGA device.[12] In this section, we describe in first and second parts respectively the CMOS sensor and the FPGA device. In the third part of this section, we present our system. Finally, in the last part, we describe briefly our image compression approach.

2.1. Image acquisition with a CMOS image sensor

We used the MTM9M413 high-speed CMOS image sensor from Micron[13] in order to design our high-speed camera. The main features of this image sensor are described as follow:

- Array format: $1,280H \times 1,024V$ (1.3 megapixels)
- Frame Rate: 500fps @ full size frame ($1,280H \times 1,024V$), $\geq$ 10,000fps @ partial size frame ($1,280H \times 128V$)
- Output: 10-bit digital through 10 parallel ports (ADC: On-chip, 10-bit column-parallel)
- Output Data Rate: 660Mbs (master clock 66MHz 500fps)
- Supply-Voltage: +3.3V
- Power Consumption: $\prec$ 500 mW @ 500 fps

2.2. Image processing with a FPGA device

The used high-speed image sensor delivers, in a pipeline dataflow mode, 500 fps with a 5Gbits per second data rate. In order to manage this dataflow, it is necessary to add inside our camera a processor able to treat in real time these informations. Some solutions are conceivable and one of them is the use of FPGA:

2.2.1. FPGA advantages for real time image processing

The bulk of low-level image processing can be split into two types of operation. The first type of operation is where one fixed-coefficient operation is performed identically on each pixel in the image. The second type of operation is neighbourhood processing, like convolution. In this case, the result that is created for each pixel location is related to a window of pixels centred at that location. These operations show that there is a high degree of repetition of processing across the entire image. This kind of processing is ideally suited to a hardware pipeline implemented in FPGA, that is able to perform the same fixed mathematical over a stream of data.

FPGAs, such as the Xilinx Virtex-II series,[5] provide a large two-dimensionnal array of logic blocks where each block contains several flip-flops and look-up tables capable of implementing many logic functions. In addition, there are also dedicated resources for multiplication and memory storage that can be used to further improve performance. Through the use of Virtex-II FPGAs, we can implement image-processing tasks at very high data rates, rates which reach hundred of MHz. These functions can be directly performed on a stream of camera data rate as it arrives without introducing any extra processing delay, significantly reducing and in some cases removing the performance bottleneck that currently exists. In particular, the more complex functions such as convolution can be mapped very successfully to FPGAs. The whole convolution process is a matrix-multiplication and as such requires several multiplications to be performed for each pixel. The exact number of multipliers that are required is dependant on the size of kernel (window) used for convolution. For a 3x3 kernel, 9 multipliers are required and for a 5x5 kernel, 25 are required. FPGAs can implement these multipliers. For example, with the one-million gates Virtex-II, 40 multipliers are available and in the eight-million gates part this number increases to 168.

2.2.2. Main features of the used FPGA

In relation with the image data rate and image resolution we choiced the VIRTEX-II XC2V3000 FPGA from Xilinx to which we present summarized specifications hereafter:

- 3,000 system gates organised in 14,336 slices
- 96 dedicated 18-bit× 18-bit multipliers blocks

- 1,728 Kbits of dual-port RAM in 18 Kbit SelectRAM resources

- 720 I/O pads

2.3. High-speed camera system

Our high speed camera system is composed of three boards as shown in figure 1. The first board contains the CMOS image sensor and is connected to the FPGA board. This second board plays three functions. The first function is the CMOS sensor control signals from the FPGA; the second function is the real time image compression data rate and the third function corresponds to real time image processing like edge detection, tracking and so on. As discussed previously, the second and third functions are also realized with the FPGA device. The role of the third board (Interface board) is to control, using the USB2 protocol, the image real time transfer between the FPGA board and the PC computer.

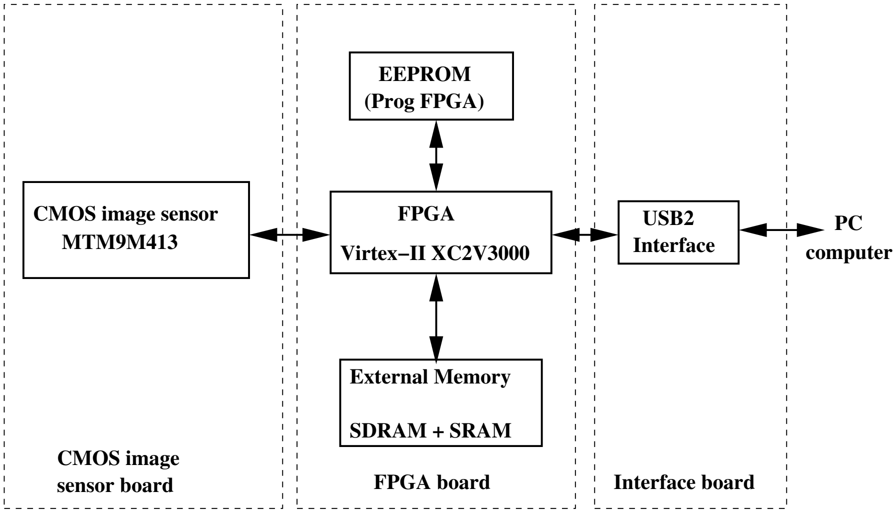

Figure 1. *High-speed camera system principle*

2.4. Embedded Image Compression Unit

The high-speed CMOS image sensor delivers images with a pixel data rate of 1024×1280 pixels $\times 500$fps $= 655$ Mpixels/s. Each pixel is coded on 10bits, thus the bit data rate is $655 \times 10 = 6.55$ Gbits/s.

As described previously, informations are sent from our high-speed camera to a PC computer through an USB2 link and this with a bit data rate range going from 340Mbits/s to 480Mbits/s. In order to send full images in real time it is necessary to compress the informations. In our case the compression ratio is $\frac{6.55 Gbits/s}{340 Mbits/s} = 19.5$.

Two main approaches are used in compression algorithms: lossless compression and lossy compression. The primary goal of lossless compression is to minimize the number of bits required to represent the original image samples without any loss of informations. All bits of each sample must be reconstructed perfectly during decompression. Some famous lossless algorithms based on error-free compression have been introduced like the Huffman coding[14] or LZW coding.[15] These algorithms are particulary useful in image archiving as in the storage of legal or medical records. These methods allow an image to be compressed and decompressed without losing information. In this case the compression ratio is low (range from 2:1 to 3:1).

Lossy compression algorithms bring higher compression ratio, typically from 10:1 to 100:1 and even more. In general more the compression ratio is high, more the image quality is low. Some famous methods, motivated for multimedia applications, have been also introduced like JPEG, JPEG2000, MPEG2, MPEG4...These methods are based on spatio-temporal algorithms and use differents approaches like predective coding, transform coding (Fourier transform, Discrete cosinus transform) or wavelet coding.

Our choice of compression method is based on two main constraints. The first concerns the real time consideration: how to compress 500 images/s in real time? The second constraint is related to the image quality, our goal in this case is to compress and to decompress images in order to obtain a PSNR* upper than 30dB. For real time consideration, compression process is realized into the FPGA device and decompression process is operated using PC computer after the image sequence has been recorded.

In our system, we combine lossless compression and lossy compression. The used lossless compression, based on Huffman algorithm, brings a compression ratio close to 2:1. So the lossy compression has to bring a compression ratio close to 10:1. In order to accomplish this target, we implemented a Wavelet coding algorithm using lifting-scheme.[16]

3. EXAMPLE OF AN HIGH-SPEED REAL TIME IMAGE APPLICATION

In the application field of fast images, in relation to time calculations limitations, real time image processing have to use only simple operations like arithmetics and logic operations, convolutions,...

*PSNR means Peak Signal to Noise Ratio and is calculated as: $PSNR = 10 log_{10} \frac{(2^B - 1)^2}{MSE}$ where B represents the number of bits in the original image. MSE means Mean Square Error and is calculated as $MSE = \frac{1}{Npixels} \sum_{x,y} (f(x,y) - g(x,y))^2$ where Npixels is the number of pixels in the image, f(x,y) and g(x,y) are respectively the grey levels of original and processed images at x,y coordinates. Reconstructed images are obtained after a compression-decompression process

In order to select some specific image processing algorithms dedicated to fast imaging, we studied some applications: real time droplet tracking, real time biomechanics analysis and real time interferogram frange position measurement. In this section, we present a specific application concerning the tracking of a mouse running on a moving pavement with a speed of 37cm/s. Here, our camera works with a 250 fps mode.

Figure 2 illustrates this application and simple image processing which can extract the X and Y positions of specific markers placed on the mouse. Thus, Figure 2-a shows the mouse running on the moving pavement. On this mouse, 7 markers have been placed. A first step of our algorithm consists to extract the ROI (Region Of Interest) where the markers are and for this, using a local edge detector, we obtain the right edge of the mouse (located near its nose). As we know the length of the mouse, we can easily determine positions of the ROI. From this ROI (shown in Figure 2-b), two image processing are executed in parallel: Image thresholding (Figure 2-c) and Edge detection using Sobel filter (Figure 2-d). Then, we merge the 3 images (Figures 2-b, 2-c, 2-d) and because some isolated white pixels appear, we erode this merged image. The final result, shown in Figure 2-e, corresponds to the markers extraction.

The final step consists to calculate, with a sub-pixel resolution, the coordinates of the centres of gravity of each detected marker.

This fast imaging application and some others (like drop tracking and interferogram frange position measurement) allow us to determine a generic scheme dedicated to fast image processing. This general scheme is illustrated in Figure 3.

4. EMBEDDED IMAGE PROCESSING BLOCKS INSIDE THE HIGH-SPEED CAMERA

From the general scheme described in the previous section (Figure 3), we have implemented inside the FPGA device of our camera (see Figure 1) some basic real time image processing operators like: ROI detection, image thresholding, image edge detection, image merging, erosion, dilation and centres of gravity calculation. These implementations have been done using VHDL language.

In particular we have previously studied the implementation of real time centres of gravity calculation in the conext of sub-pixel metrology.[17] We used this result for our application. In order to illustrate our approach, we present in this section two descriptions of real time image processing implementations: Edge detection using Sobel filter and Erosion-Dilation.

4.1. Real time Edge detection using Sobel filter

If we consider $f(x, y)$ the grey value of the current pixel, where x and y are the pixel coordinates, the output $g(x, y)$ of the Sobel filter edge detector is given by the following equation: $g(x, y) =$

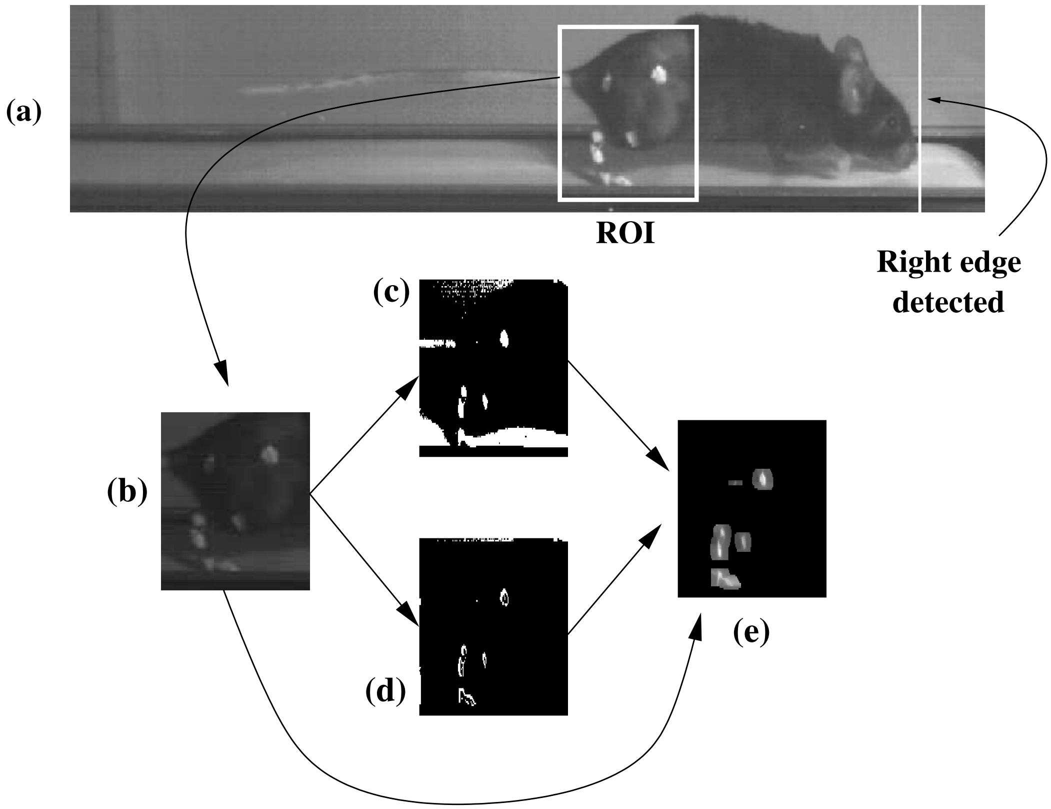

Figure 2. *Mouse tracking*

$|(g_1(x,y))| + |(g_2(x,y))|$ where:
$g_1(x,y) = (f(x,y) + 2f(x-1,y) + f(x-2,y)) - (f(x,y-2) + 2f(x-1,y-2) + f(x-2,y-2))$ and
$g_2(x,y) = (f(x,y) + 2f(x,y-1) + f(x,y-2)) - (f(x-2,y) + 2f(x-2,y-1) + f(x-2,y-2))$.
Using the Z transform respectively to $g_1(x,y)$ and $g_2(x,y)$, we obtain $G_1(z)$ and $G_2(z)$. Thus, $G_1(z) = (F(z) + 2Z^{-1}F(Z) + Z^{-2}F(Z)) - Z^{-2N}(F(z) + 2Z^{-1}F(Z) + Z^{-2}F(Z))$ where $F(z)$ is the Z transform of $f(x,y)$ and N is the number of pixels per line.
$G_1(z)$ can be factorized as: $G_1(z) = (F(z).(1+Z^{-1}).(1+Z^{-1})) - Z^{-2N}(F(z).(1+Z^{-1}).(1+Z^{-1}))$. This last equation shows that the $g_1(x,y)$ component can be calculated using simple first order digital FIR filters. Similarely, we obtain the $g_2(x,y)$ component and the full schematics of the Sobel filter operator is given in Figure 4. Using this implementation, the time calculation with the FPGA device is less than 20ns per pixel.

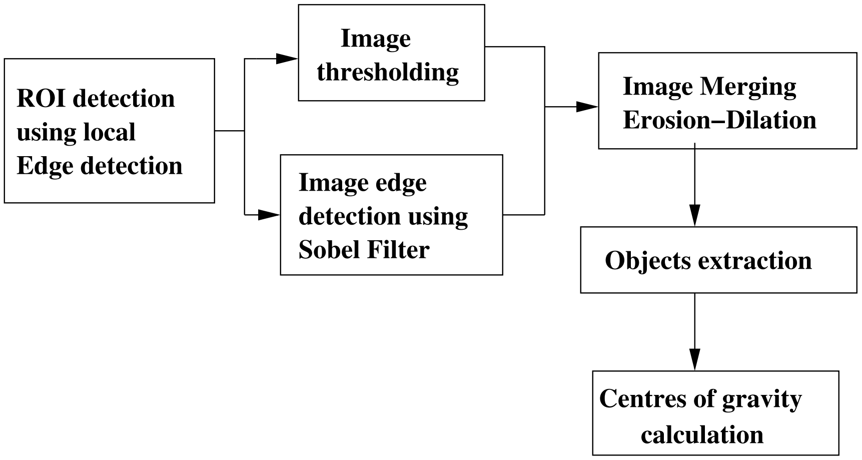
ROI detection using local Edge detection
Image thresholding
Image edge detection using Sobel Filter
Image Merging Erosion−Dilation
Objects extraction
Centres of gravity calculation

Figure 3. *Generic Image processing*

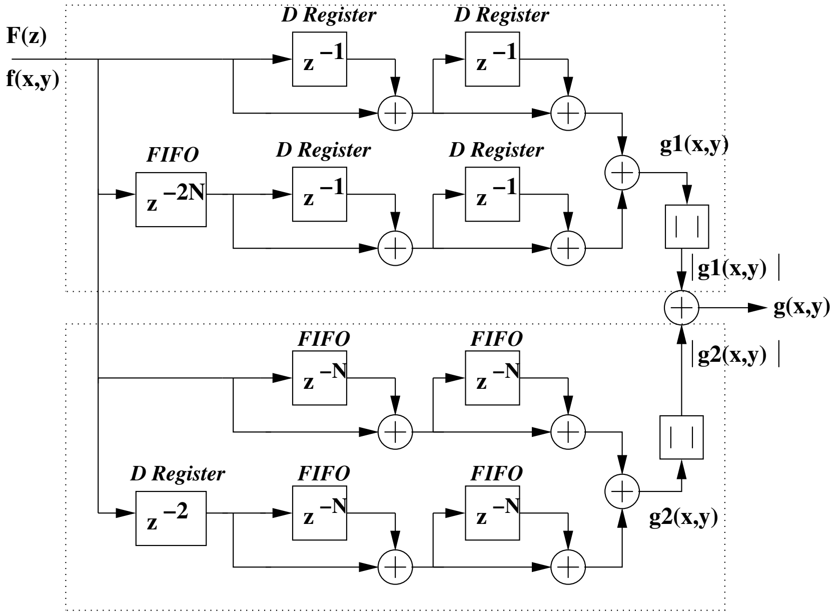

Figure 4. *schematics of Sobel calculation*

4.2. Real time Edge Erosion and Dilation

Erosion and Dilation operations correspond to calculate respectively the minimum and the maximum values inside a window and to replace the current pixel value by these new values. Thus, for a 3×3 window Erosion and Dilation are obtained as follow:

$erosion(x, y) = minimum(f(x, y), f(x-1, y), ..., (f(x-2, y-2))$ and
$dilation(x, y) = maximum(f(x, y), f(x-1, y), ..., (f(x-2, y-2))$. One of the solutions to implement these two operations is illustrated in Figure 5. Again, the time calculation with the FPGA device is less than 20ns per pixel.

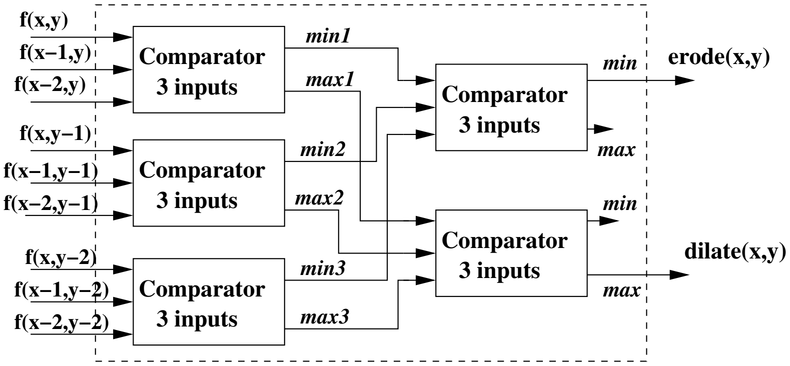

Figure 5. *schematics of Erosion and Dilation calculation*

5. NEW DEVELOPMENTS OF FAST EMBEDDED IMAGE PROCESSING

In order to improve performances of our high speed camera, we conducted two developments that we describe hereafter:

5.1. New concept of fast smart camera

In general, the readout capabilities of vision chips are limited. Although they perform their computation in an SIMD scheme, they often send pre-processed data sequentialy. This is an important bottleneck in the datapath and reduce considerably the available bandwidth.

An other remark we can make is that there are few pertinent data in an image. In image computing, most applications work on objects of interest, e.g. Data-Matrix, land-mark detection, people tracking, fingerprint recognition. Consequently why process the remainder of the image?

In order to answer these questions, we choose to use a sensor with Region Of Interest (ROI) readout and parallel outputs capabilities. It is linked to RAM or an array of processors that are able to store/compute provided data in order to provide an efficient way of acquiring and computing ROIs. Our system has been described in details in Heyrman and al[18] and is represented in figure (6). In order to validate our new concept we studied some applications like movement analysis. Using a standard architecture based on an ARM processor connected to a FPGA, the computation time for

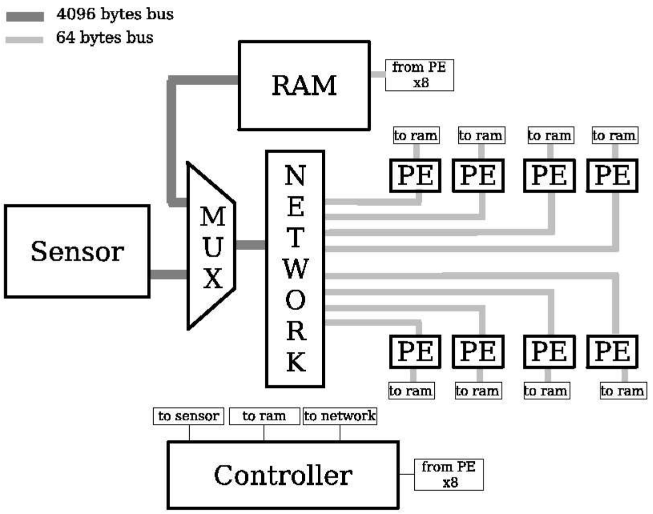

Figure 6. *Fast smart sensor architecture*

this application is $40ms$. Using our new architecture, the computation time is $36\mu s$, thus 1000 times faster.

5.2. New high speed CMOS image sensor with embedded analog image processing

In order, to improve the fast image sensor performances, we designed a new high speed analog VLSI image acquisition and pre-processing system. This ASIC has been fabricated in the 0.35 μm standard CMOS process from AMS. The chip features a massively parallel architecture enabling the computation of programmable low level image processing in each pixel. Extraction of spatial gradients and convolutions such as Sobel filter or Laplacian are implemented on the circuit. For this purpose, each pixel of 35 $\mu m \times 35\mu m$ includes a photodiode, an amplifier, two storage capacitors and an analog arithmetic unit based on a four-quadrants multipliers architecture. The retina provides address-event coded output on three asynchronous buses, one output is dedicated to the gradient and both others to the pixel values.

As shown in figure 7, a proof-of-concept chip of 64 x 64 pixels was fabricated. A dedicated embedded platform including FPGA, ADCs has also been designed to evaluate the vision chip.

Measured results show that the proposed sensor successfully captures raw images up to 10 000 frames per second and runs low level image processing at a frame rate comprised 2 000 and 5 000 frames per second. A full description of this new vision chip is done in Dubois and al.[19]

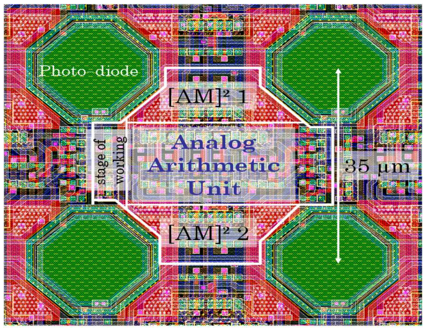

Figure 7. *Fast analog vision chip*

6. CONCLUSION AND PERSPECTIVES

In this article we have shown that it is possible to implement into a FPGA device several real time image processing for some specific fast imaging applications like real time droplet tracking, real time biomechanics analysis and real time interferogram frange position measurement. In particular, we have described an application concerning embedded tracking of a mouse running on a moving pavement with a speed of 37cm/s. For this application basic image processing, like edge detection, erosion and centres of gravity calculation have been studied and implemented into the FPGA device. Our recent developments have shown that it is possible to capture and to process fast images respectively with a frequency of 10Kfps and 5Kfps.

In perspectives, we wish to design and to implement new algorithms for Fast Motion capture without markers using pattern recognition approaches like Neural Networks and for this, we will use previous results that we already obtained for embedded real time face tracking.[20]

REFERENCES

1. E.Fauvet, M.Paindavoine, F.Cannard *Fast image analysis system*, 19th International Congress on High-Speed Photography & Photonics Cambridge (UK) - Septembre 1990.
2. F. Bouffault, C.Milan, M.Paindavoine, J. Febvre, G. Cathebras, *High speed cameras using CCD image sensor and new high speed sensor for biological applications*, 21th International Congress on High-Speed Photography & Photonics, Taejon , September 1994.

3. F.Bouffault, J.Febvre, C.Milan, M.Paindavoine, J.C.Grapin *High speed video microsystem*, Measurement Science and Technology Journal (IOP Publishing), vol. 8, n5, pp. 398-402, may 1997.
4. M.Paindavoine, D.Dolard, JC.Grapin, *Real-time imaging system applied to human movement analysis*, SPIE International Symposium on Optical Science and Technology, Denver, USA, July 1999
5. Xilinx Company, *FPGAs devices*, http://www.xilinx.com
6. Vision Research Company, *Phantom CMOS 1600 × 1200 Camera*, http://www.visiblesolutions.com/phantomv9.htm
7. Nac Company, *Memrecam fx K4 High-Speed Color Video System*, http://www.nacinc.com
8. Photron Company, *Ultima APX-RS Fastcam*, http://www.photron.com
9. Redlake Company, *MotionXtra high speed system*, htpp://www.redlake.com
10. Optronis Company, *CamRecord 600 high speed camera*, http://www.optronis.com
11. Weinberger Company, htpp://www.weinbergervision.com
12. M.Paindavoine, R.Mosqueron, J.Dubois, C.Clerc, JC.Grapin, L.Pierrefeu, F.Tomasini, *High-speed camera with internal real-time image processing*, 26th SPIE High Speed Photography and Photonics Conference, Alexandria-USA, September 2004
13. Micron Company, *MT9M413 CMOS Image Sensor*, http://micron.com/products/imaging/
14. D.A.Huffman, *A method for the construction of minimum redundancy codes*, Proc.IRE 1952, vol.40, no10, pp1098-1101
15. J.Ziv, A.Lempel, *A universal algorithm for sequential data compression*, IEEE Transactions on information theory, 1977
16. W.Sweldens, *The lifting scheme: A new philosophy in biorthogonal wavelet constructions*, Wavelet applications in signal and image processing III, pp.68-79, Proc. SPIE 2569, 1995
17. D.Rivero, M.Paindavoine, S.Petit, *Real-Time Sub-pixel Cross Bar Position Metrology*, Real-Time Imaging Journal, vol.8 pp 105-113, 2002
18. B.Heyrman, M.Paindavoine, R.Schmit, L.Letellier, T.Collette, *Smart camera design for intensive embedded computing*, Real-time Imaging Journal, 11, pp. 282-289, 2005.
19. J.Dubois, D.Ginhac, M.Paindavoine, *VLSI Design of a High-Speed CMOS Image Sensor with in-situ 2D Programmable Processing*, EUSIPCO 2006, Florence, ITALY, 8 September 2006.
20. F.Yang, M.Paindavoine, *Implementation of a RBF neural network on embedded systems: Real time face tracking and identity verification*, IEEE Transactions on Neural Networks, Vol.14, N5 pp. 1162-1275, September 2003.