Hardware Architecture for Visual Feature Extraction

Merwan BIREM[1] and Francois BERRY[2]

***Abstract*— This paper proposes a real-time architecture for image feature extraction based on the Harris corner detector algorithm. This architecture has been developed on the Dream-Cam, which is is a modular smart camera constructed with the use of a Field Programmable Gate Array (FPGA) like main processing board. In this approach, a pixel stream is read directly from a CMOS image sensor and produces as output the detected features. Each feature is identified by its coordinates (x,y) and described by its neighborhood. In order to speed up the processing and to avoid the iterative form of the initial algorithm, the steps are "unrolled" into a sequence of identical computation blocks arranged in a pipeline architecture. Experimental results demonstrate the accuracy and robustness of the proposed system.**

***Index Terms*— Image Processing - Visual Feature Extraction - VHDL - Harris & Stephen - Field Programmable Gate Array (FPGA) - Hardware Implementation - Real-Time System.**

I. Introduction

One of the fundamental tasks of an autonomous robot is to build a map of the environment and use it for self localization. The problem of Simultaneous Localization and Mapping (SLAM) has therefore received much attention in the last decade [11]. Nowadays, approaches using laser range finders are very successful. SLAM using vision, however, remains a challenging research topic, e.g [5]. Using a camera has the advantage over a laser-range finder that it is a passive sensor that is low cost, low power, and lightweight. A camera furthermore provides a rich source of information, which enables the use of sophisticated detection and recognition methods. The difficulty, however, is to extract relevant features from the highdimensional visual data in real time.

To overcome this difficulty we propose to use instead of traditional camera a smart camera which has it's own processing unit. In our study the processing unit of the Dream-Cam is an FPGA, and to validate the architecture of the Dream-Cam we implement on it a feature extractor which is the combination of the Harris corner detector [7] and a simple descriptor. An overview of the state of the art about feature extractors is given by Mikolajczyk and Schmid [8]. The Harris corner detector (also known as Harris-Stephens or Plessy detector) is one of the most widely used interest point detectors, thanks to its improved detection rate over the Moravec [9] corner detector and his high repeatability rate [10].

*This work reported in this paper was supported by European Program Euripides (Eureka) - Seamoves Project - and the Altera Corporation under an equipment grant.

[1]M. BIREM is a PhD student in Electronics and System Architecture, Balise Pascal University, 63000 Clermont-Ferrand, France merwan.birem at etudiant.univ-bpclermont.fr

[2]F. Berry is Associate professor, Balise Pascal University, 63000 Clermont-Ferrand, France berry at univ-bpclermont.fr

The paper is organized as follows. In section II the smart camera *Dream-Cam* is presented, and in section III our own hardware orientated architecture of the interest points detector algorithm is explained. After that, experimental results are given in order to verify the system functionality and performance in section IV. Finally, a conclusion is made in section V.

II. Hardware Description of the "DreamCam"

Our approach consists in integrating the control of the imager in the perception loop, especially in the early perception processes. By integration of early processing, close to the imager, a reactive sensor can be designed. With such a smart sensor, it is possible to perform basic processing and the selection of relevant features [4].

Most vision applications are focused on several small image areas and consequently the acquisition of the whole image is not always necessary. It is evident, therefore, that one of the main goals of an efficient vision sensor is to select windows of interest (WOI) in the image and concentrate on processing resources on these. The notion of local study is then predominant and the choice of imaging technology becomes crucial.

A. Global Hardware Architecture

The global processing system is composed of System on Programmable Chip (SOPC), by which an entire system of components is put on a single chip (FPGA). Fig-1 gives an overview of the embedded vision system.

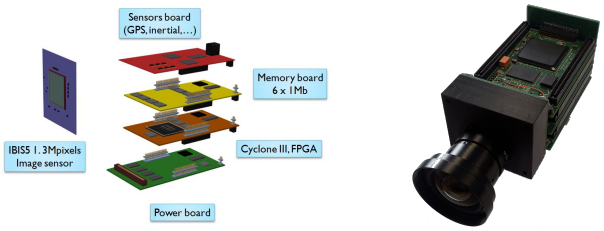

Fig. 1. Overview of the camera and synoptic of the assembly

The DreamCam is composed essentially of three boards :

1) Sensing device board: CCD and CMOS are the two most common imaging technologies used today in industrial digital cameras. A particularity of CMOS imagers is to adopt a digital memory style readout, using row decoders and column amplifiers. Random access of pixel values is possible and allows the selection of readout of WOI. The sensing device board is composed of a CMOS imager (IBIS5-A-1300) manufactured by Cypress. This 1.3-mega pixel (1280×1024) CMOS active pixel sensor dedicated to

Image Resolution (X*Y)	Frame rate (frames/s)	Frame readout time (ms)	Comment
1280 × 1024	27.5	36	Full resolution
640 × 480	100	10	ROI readout
100 × 100	1657	0.6	ROI readout

TABLE I
FRAME RATE VS RESOLUTION [3]

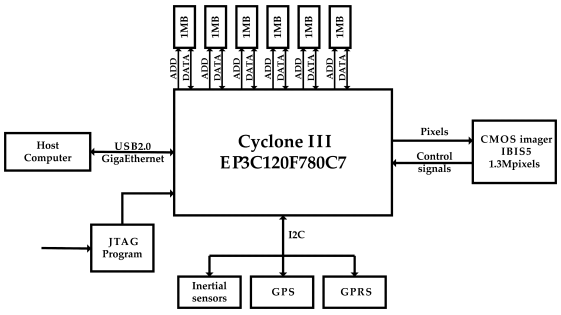

Fig. 2. Global synoptic of the sensor

industrial vision features both rolling and snapshot shutter. Full frame readout time is 36ms (max. 27.5 fps), and readout speed can be boosted by windowed region of interest (ROI) readout (Table-I).

2) Processing board: This is the core of the system designed around an Altera Cyclone-III EP3C120 (Fig-2). The need for strong parallelization led us to connect 6×1MBytes SRAM asynchronous memory blocks. Each memory has a private data and address buses. Consequently, in the FPGA, six processes (using 1MB each) can address all the memory at the same time.

3) Communication board: This last board is connected to the main board and manages all communications with the host computer.

B. Architectural Design

The aim of the proposed design is to create a flexible interface between the sensing device board and the host computer. The approach chosen is based on several HDL blocs that control the imager, the USB and the memories (all at the same time).

The system can be separated into four parts : CMOS image acquisition module, USB data transmission module, RAM data module, and image data acquisition and display module. The diagram of the system is shown in Fig-3.

After powering the system, the CMOS imager starts to work under the control of FPGA.

III. HARRIS CORNER EXTRACTOR APPLICATION

In an image, the corner is an important local feature which focuses on a great amount of important image information and is rarely affected by illumination change. In addition, it has rotation invariant properties. Thus, corner detecting has many important applications in practice, especially in the real-time target tracking field and autonomous navigation of vehicles.

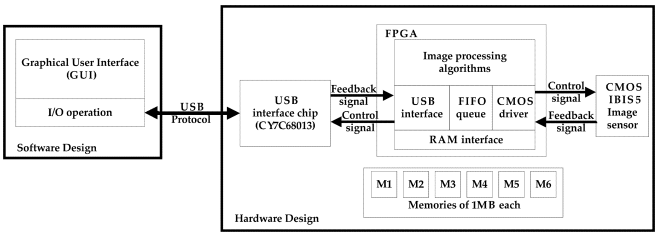

Fig. 3. System Working Diagram

In the Harris corner detector the main operations used are the first derivative and the convolution. These operations are single instruction multiple data (SIMD) and therefore highly parallelizable, which means they are suitable for implementation on FPGAs, which are low cost and high density gate arrays capable of performing many complex computations in parallel while hosted by conventional computer hardware.

The feature extraction system proposed in this paper detects first the interest points on an image, sorts them, and then describes them using a patch of pixels from the image. The system is composed of several modules (see Fig-4) that have been developed in VHDL (VHSIC Hardware Description Language), and are fully compatible with a FPGA implementation. The main modules of the system are as follows.

1) The Harris corner detector module : detects the interest points and filters them, in order to obtain only one point (pixel) for each corner,
2) The sort module : sorts the interest points in decreasing order according to their interest value R_i,
3) The swap memories module : retrieves the patch of each interest points, and constructs the data frame containing the interest point coordinates (X_i, Y_i) and their patches. More details about the data frame are given in section III-C.

In the first module the results of the detection are filtered, because when an image is treated using the Harris corner detector, several pixels around a corner will be considered as interest points. The desired outcome is having one interest point, which means one pixel for each corner (see Fig-6). In previous works [1], [2], [6], this problem was solved by the non-maximum suppression method. To perform such a suppression, a window of a specific odd size is moved over the picture after treating all the pixels. If the center value of the window is the maximum interest value within the whole window the filter response is one, otherwise the filter response will be zero. In terms of hardware considerations, this method has several disadvantages such as the use of more memories and FIFO queue. In addition, it induces latency due to the buffering of three lines at minimum when a 3×3 window is used to perform the non-maximum suppression.

A. Harris corner detector module

This module represents the feature detector (the first part of the feature extractor). Fig-5 gives an overview of the

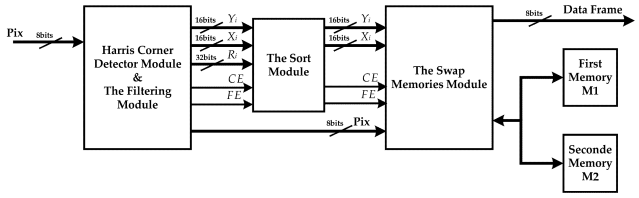

Fig. 4. Block diagram of the implemented system, composed of three hardware blocks, used to extract features from the images

architecture used to implement the Harris detector algorithm on FPGA.

To achieve a higher frequency, the system has to be parallelized to a maximum degree. As shown in Fig-5, all operations that are independent of one another were implemented separately. The performance of the system can also be increased by using DSP-blocks for all the multiplications and summations that are in the Harris corner detector algorithm.

The system receives the stream of pixels and places them one by one in a FIFO queue. The computation of the interest value R_i will start when the FIFO queue is almost full, more precisely when the second pixel of the fifth line is reached. After the computation of the interest value R_i, a simple comparator is used to determine if the treated pixel is an interest point or not.

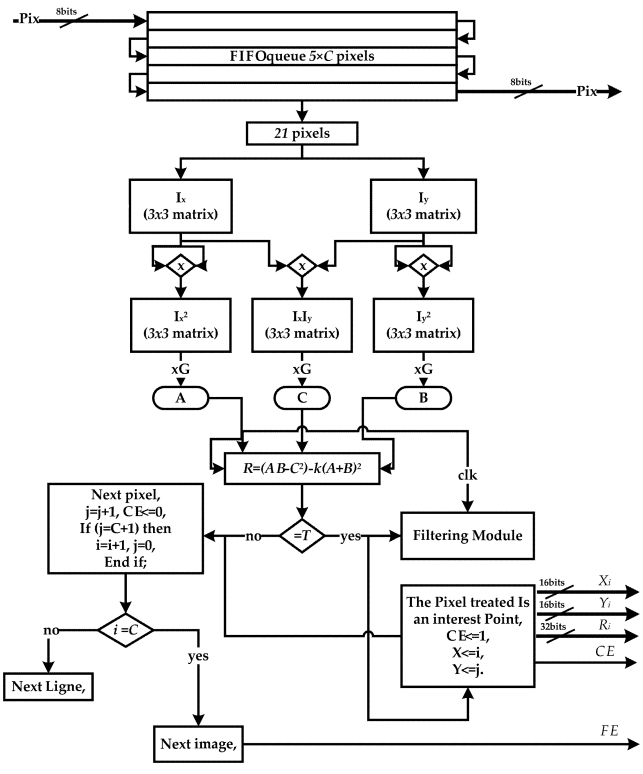

Fig. 5. Block diagram of the implemented Harris corner detector module

This module contains a submodule that has nearly the same function as the non-maximum suppression. The main difference between the two is in the pixel that will be kept. In the non-maximum suppression the pixel kept is the one with the highest interest value R_i in an odd-size window. In the submodule implemented on FPGA the pixel kept is the first one to appear, which means there is no need for more memory or latency to obtain the results.

This submodule is based on two notions. When an interest point is detected the system will check if there is no interest point near it in the m previous lines. If this is the case, the pixel will be set as an interest point, and the n following pixels will not be treated at all. If there is an interest point near it in one of the m previous lines the system passes to the next pixel and so on. Fig-6 shows an example of the results obtained with and without this module.

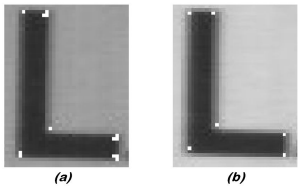

Fig. 6. An example of the results obtained with the filtering module (real-time images) : (a) Detection without filtering, (b) Detection with filtering

The most important output signals of this module are CE, FE, X_i and Y_i. The first and second ones are set to "1" (during one clock cycle) when an interest point is detected and when the last pixel of an image is treated, respectively. The last two signals, X_i and Y_i, represent the coordinates of the detected interest point.

B. Sort module

The robustness and accuracy of an interest point depends on the value of R_i : the higher the value of R_i, the more robust and accurate the point.

The technique used to sort the interest points is that described in the paper of [12]. It is done in two steps :

- ***Step 1 :*** The ordering process, in which the order of the detected interest points is found,
- ***Step 2 :*** The rearranging process of the points, which places them in a memory according to their order.

The principle of this sorting method is as follows. For a given set of interest points $SetIP = \{IP_1, IP_2, \ldots, IP_n\}$, the order C_i (of each interest point) is easily computed by counting results of comparisons between R_i (the interest value of the i^{th} point) and all the other values. Each time a value higher than R_i is found, C_i is incremented by 1. C_i represents the number of items in the set having a value higher than the i^{th} point, and represents the order of the point. The rearranging process uses the different value of C_i as addresses to put the points in decreasing order in a memory.

The basic algorithm to compute the order C_i is shown in Algorithm 1.

C. Swap memory module

This module represents the feature descriptor (the second part of the feature extractor). As mentioned previously, the

Algorithm 1 Compute the order : C_i

$C_i = 0$
while $j \leq n$ **do**
  if $R_i < R_j$ **then**
    $C_i = C_i + 1$
  end if
  $j = j + 1$
end while

descriptor chosen to be implemented on FPGA is a simple one, which gives for each interest point an intensity patch. This module receives signals CE, FE, X_i and Y_i from the sort module. The signals are used to construct the data frame shown in Fig-7.

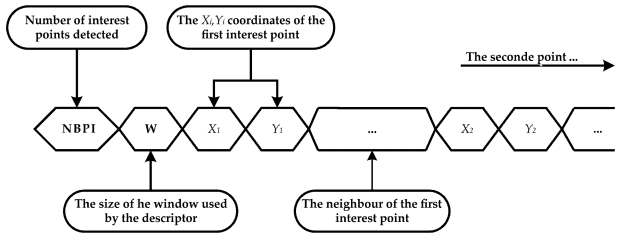

Fig. 7. The data frame constructed by the swap memory module

The first two elements and all the (X_i, Y_i) coordinates of the data frame are obtained from the Harris corner detector and sort modules. The pixels that are in each patch are obtained from one of the two memories, $M1$ or $M2$ (see Fig-4). These memories contain the previous treated image (put on read mode), and the actual image under treatment (put on write mode), respectively. At the end of each image treatment the two memories change their operating mode i.e. swap mode.

This module is composed of two processes. The first one constructs the table that will contain the memory addresses of the detected interest points. The second process constructs the data frame. To do this, the process uses the memory put on read mode and the table constructed by the first process.

The combination of this module with the Harris corner detector and the sort module will give us a full feature extractor that detects, sorts, and describes the interest points. In other words, the extractor takes images as input and provides semantic information as output. This information can be used for navigation, 3D reconstruction or other applications.

IV. Experimental results

The proposed system (presented in the previous section) was implemented on the Dream-Cam. Image resolution of 800×1024 was used and the size of each interest point patch was set at 15×15. The hardware blocks were developed to process up to $33.8M$ pixels per second, which corresponds to $41.18 fps$ of 800×1024 pixels each.

The operational maximal frequency obtained for the Harris corner extractor block on the Cyclone-III was equal to 56.69MHz, but because we were working in the flow of pixels coming from the imager directly, the frequency of IBIS sensor -which is equal to 40MHz- had to be used (this is why the system works at the same frame rate as the sensor). Table-II presents the resources used to implement the whole system (IBIS controller blocks, USB blocks and Harris features extractor blocks) on the Altera Cyclone-III, for the following project : image size = $800 \times 1024 pixels$, patch size = $15 \times 15 pixels$, and greatest number of interest points = 100.

Total Logic Element	18'539 / 119'088 (16%)
Total Pins	296 / 532 (56%)
Total Memory Bits	1'121'520 / 3'981'312 (28%)
DSP Block 9-bit elements	33 / 576 (6%)
Total PLLs	1 / 4 (25%)

TABLE II
FPGA RESOURCES USED

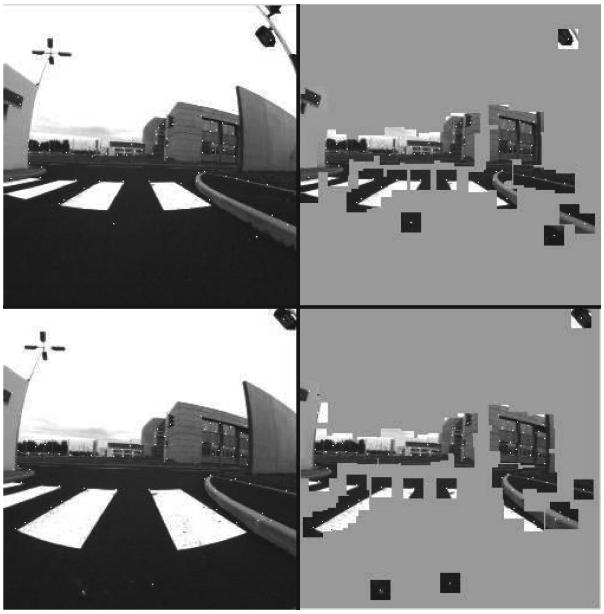

Fig. 8. The results obtained when processing images were captured from a mobile robot. On the left the whole image with the interest points in white. On the right the point of interest and its neighbor

Fig-8 shows the results obtained when processing images captured from a mobile robot in experimental conditions. Images on the left are obtained without the feature descriptor, and images on the right are obtained with the feature descriptor. The latter are constructed using the data frame that the Harris corner extractor sends to the host PC.

The Harris corner extractor implemented on FPGA works on the stream of pixels coming from the CMOS imager and because of this the size of the data frame must be lower or equal to that of the image treated to allow the data frame to be sent entirely without the loss of any information. In general, if the images treated have $L \times C pixels$ each, and the patch chosen to describe the interest points has $W \times W pixels$, then the maximal number of interest points that the system can detect is $n < \frac{L \times C - 2}{W \times W + 4}$, where -2 is for the first two elements

of the data frame (the number of interest points detected, and the size of the patch), and $+4$ is for the two coordinates X_i, Y_i of each interest point. The $+4$ means that the two coordinates are encoded on $2bytes$ each, which will allow the system to obtain the coordinate of interest points detected in images that have a size greater than $256 \times 256 pixels$.

V. Conclusion and Future Work

This paper describes the construction of a sensor for real-time visual applications. Its main originality consists in using CMOS imager and FPGA architecture to create a versatile smart system. The approach, based on FPGA Technology and CMOS imager, reduces the classic bottleneck between sensor and processing unit. ROI readout guarantees the high frame rate of the system (more than $100fps$ for 640×480 pixels). The average transmission speed with USB is 48MB/s, which will meet the demands of real-time data transmission. The system can be used in many applications with demands of high resolution, high frame rate and real-time requirements. The feature extractor application was implemented with success on the Dream-Cam, which can process up to $42fps$ (800×1024 pixels) and gives good quality results, as seen in section IV. The blocks of this application were developed in generic mode, which means the user can change the size of the image, the number of points needed, the threshold value, and other parameters, compile, and synthesize the project to obtain a new system.

References

[1] Dietrich Benedikt. *Design and Implementation of an FPGA-based Stereo Vision System for the EyeBot M6*. PhD thesis, University of Western Australia, 2009.

[2] Metz Brandon and Morfopoulos Arin. Rapid corner detection using fpgas. *National Aeronautics and Space Administration*, December 2010.

[3] Semiconductor Corporation Cypress. *IBIS5-A-1300 1.3M Pixel Dual Shutter Mode CMOS Image Sensor Datasheet*, 2005.

[4] André DeHon. The density advantage of configurable computing. *Computer*, 33(4) :41–49, April 2000.

[5] Simone Frintrop and Patric Jensfelt. Attentional landmarks and active gaze control for visual SLAM. In *IEEE Transactions on Robotics, special Issue on Visual SLAM*, volume 24, October 2008.

[6] Deborah Goshorn, Junguk Cho, Ryan Kastner, and Shahnam Mirzaei. Field programmable gate array implementation of parts-based object detection for real time video applications. Washington, DC, USA, 2010. IEEE Computer Society.

[7] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988.

[8] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal*, 27 :1615–1630, October 2005.

[9] Hans Moravec. Obstacle avoidance and navigation in the real world by a seeing robot rover. In *tech. report CMU-RI-TR-80-03, Robotics Institute, Stanford University*. September 1980.

[10] Oscar Martinez Mozos, Arturo Gil, Monica Ballesta, and Oscar Reinoso. Interest point detectors for visual slam. In *Proceedings of the Conference of the Spanish Association for Artificial Intelligence*, Salamanca, Spain, 2007.

[11] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[12] H. Yasuura, N. Takagi, and S. Yajima. The parallel enumeration sorting scheme for vlsi. *IEEE Trans. Comput.*, 31 :1192–1201, December 1982.