Locust-Inspired Vision System on Chip Architecture for Collision Detection in Automotive Applications

Luis Carranza, Rubén Laviana, Sonia Vargas, Jorge Cuadri, Gustavo Liñán,
Elisenda Roca and Angel Rodríguez-Vázquez
Instituto de Microelectrónica de Sevilla, CNM-CSIC
Sevilla, Spain
carranza@imse.cnm.es

Abstract— This paper describes a programmable digital computing architecture dedicated to process information in accordance to the organization and operating principles of the four-layer neuron structure encountered at the visual system of Locusts. This architecture takes advantage of the natural collision detection skills of locusts and is capable of processing images and ascertaining collision threats in real-time automotive scenarios. In addition to the Locust features, the architecture embeds a Topological Feature Estimator module to identify and classify objects in collision course.

I. Introduction

Machine vision represents an exciting and dynamic part of cognitive and computer science. Through the combination of advanced hardware and software technologies, artificial vision is ready to confront challenges related to highly demanding sensory/processing/actuating applications [1]. One of these challenges is collision detection in automotive applications, which actually defines an active research field in the automotive sector [2].

In this paper we propose a System on Chip (SoC) computing architecture for collision detection in automotive scenarios (figure 1). This design is capable of processing visual information according to the four-layer natural neural network found in the visual system of locusts [3], and includes a Topological Feature Estimator (TpFE) resource for object identification. The system comprises a general purpose processor, which plays a master role, and arithmetic circuitry resources devoted to the number-crunching and memory accessing tasks involved in the locust neuron's model and TpFE calculations. These tasks are guided and multiplexed in time by software. The input images are acquired using a High Dynamic Range (HDR) CMOS image sensor which is low-level controlled with a full custom RISC microcontroller (LμC) whose instruction set and architecture have been optimized for efficient control functions and low area consumption.

The system comprises three levels of control hierarchy; the general-purpose processor at the top-level, the locust neurons model and feature estimator arithmetic computing units together with the RISC microcontroller at the second, and the HDR CMOS image sensor at the third.

Software changes and improvements can be easily made as the general purpose processor is released from the duties related to control, arithmetic and memory accessing processes involved in collision threat detection, feature estimation and low level HDR sensor related control tasks. The system's digital core will first be synthesized on an FPGA and after being tested and fully verified, included together with the HDR sensor within the same ASIC.

The locust-inspired neural network model is presented in Section II. The TpFE's main ideas are discussed in Section III. Section IV shows a general view of the full custom computing units. The HDR Sensor and the LμC are described in Sections V and VI, and the conclusions are summarized in Section VII.

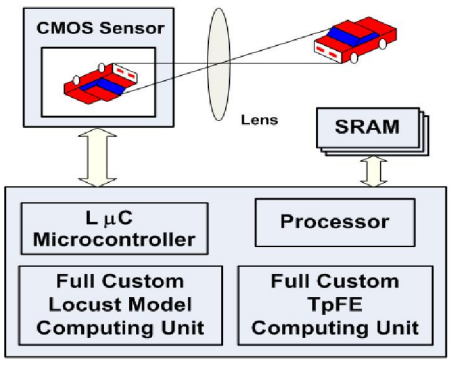

Figure 1. Block diagram of the proposed system on chip architecture.

II. Bioinspired Model

The collision threat detection model, inspired by the visual system of locusts, has been adapted and tuned in successive progressive refinement steps for VLSI integration [4][5]. The inputs to the network are external images and the outputs are collision threat signals coming from the latest layer, comprised of a single neuron, the lobula giant movement

detector (LGMD). Figure 2 shows a block diagram of the locust vision system neural network model, where every neuron layer extracts information from the previous one in a destructive way performing the following operations:

Layer 1 (L1). The input to the neural network is the luminance of external scenes. P cells are arranged in a 150x100 bidimensional array of photoelectric detectors which transduce light to an electric signal L_{ij} representing the captured external scene.

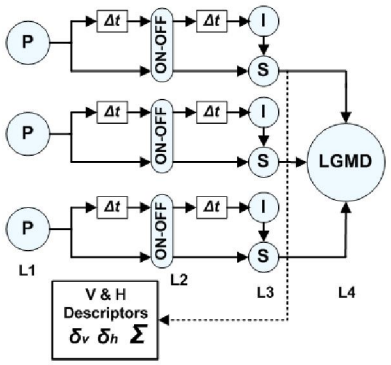

Figure 2. Four-layered locust vision system neural network model together with the horizontal and vertical statistical descriptors.

Layer 2 (L2). ON-OFF neurons isolate the moving object from the background, calculating a movement map, i.e. the difference between the current and the previous frame:

$$C_{ij}(t) = \left| \frac{L_{ij}(t) - L_{ij}(t - \Delta t)}{2} \right| \qquad (2)$$

Layer 3 (L3). The movement map signal is transmitted to the inhibitory *I* cells, where inhibition spreads to neighboring cells, being delayed before interacting with the current excitation. This process removes much of the excitation caused by small changes in background movement, which may cause spurious alerts. The signal coming from the ON-OFF neurons is scaled in the inhibition neurons:

$$I_{ij}(t) = Incoeff(n) \cdot C_{ij}(t - \Delta t) \qquad (3)$$

where the coefficient depends on the number of spikes generated by the LGMD (layer 4) within a predefined time window (5 frames in our case).

S-units perform a half wave rectification of the competition result of signals coming from ON-OFF cells and signals coming from inhibition *I*-units, generating a net activity potential as:

$$S_{ij}(t) = Max(C_{ij}(t) - I_{ij}(t), 0) \qquad (4)$$

Layer (L4). Within the LGMD neuron, the global excitation is obtained, taking into account the information generated by all *S* cells:

$$e(t) = \frac{\sum_{i,j} S_{ij}}{150} \qquad (5)$$

The membrane potential of this neuron is the solution of the following z-domain temporal differential equation:

$$\begin{aligned} LGMD(z) &= \alpha_2 z^{-2} LGMD(z) \\ &+ \alpha_1 z^{-1} LGMD(z) + \alpha_0 e(z) \end{aligned} \qquad (6)$$

The LGMD neuron fires a spike if its membrane potential exceeds an adaptative threshold, given by:

$$V(z) = \beta_3 z^{-14} v(z) + \beta_2 z^{-10} v(z) + \beta_1 z^{-5} v(z) + \beta_0 \qquad (7)$$

where $\beta_3, \beta_2, \beta_1, \beta_0$ are adjustable coefficients and $v(z)$ is the solution of the equation:

$$v(z) = \alpha_2 z^{-2} v(z) + \alpha_1 z^{-1} v(z) + \alpha_0 e(z) \qquad (8)$$

If during 5 consecutive frames the LGMD generates at least 4 spikes, the collision threat detection alarm is fired.

III. Topological Feature Estimator

The object classification task is carried out using the object's shape information filtered by *S* cells (figure 2), taking advantage of the fact that the information present in the *S* cell layer is free from background and irrelevant items and only the movement map of the main objects in the field of view remains. Consequently, it is particularly appropriate for extracting information about the geometrical features of the moving objects that are in the field of view.

Consider the sum of every row ($v_i(t)$) and column ($h_j(t)$) of the *S* cell layer output:

$$v_i(t) = \sum_{j=1}^{150} S_{ij}(t) \quad , \quad h_j(t) = \sum_{i=1}^{100} S_{ij}(t) \qquad (9)$$

Being, respectively, $\overline{v(t)}$, $\overline{h(t)}$, $\sigma_v(t)$, $\sigma_h(t)$ their mean value and their variance, the following descriptors can be defined as:

$$\delta_v = \overline{v(t)} - \sigma_v(t) \quad \text{vertical deviation} \qquad (10)$$

$$\delta_h = \overline{h(t)} - \sigma_h(t) \quad \text{horizontal deviation} \qquad (11)$$

With this information it is possible to perform the following elementary object classification:

1. Vertical-shaped objects (pedestrians, traffic lights):

$$\delta_v(t) > 0 \ \ \& \ \ \delta_h(t) < 0 \qquad (12)$$

2. Horizontal-shaped objects (road stripes…):

$$\delta_v(t) < 0 \ \ \& \ \ \delta_h(t) > 0 \qquad (13)$$

3. Rectangular-shaped objects (vehicles, obstacles…):

$$\delta_v(t) > 0 \;\&\; \delta_h(t) > 0 \;\&$$
$$\&\; \left[\left(Max\{h_j(t)\}\right) - \overline{h(t)}\right] > a \;\&$$
$$\&\; \left[\left(Max\{v_i(t)\}\right) - \overline{v(t)}\right] > b \qquad (14)$$

Where a and b are adjustable parameters that depend on the dynamic range of the pixels in the S layer and the expected dimensions of the objects to be classified [4].

If none of the cases occur, the object remains unidentifiable.

Experimental results have been obtained processing, at real time, video traffic scenes provided by Volvo Car Corporation, according to a software implementation of the collision threat detector model, which fires an alarm when an object is in collision course, and the TpFE model, whose main job is to identify the object that is firing this alarm. This software implementation, which runs on a desktop PC, shows that the TpFE classification procedure identifies correctly about the 73% of the objects that are firing the collision alarm [4].

IV. Full Custom Computing Units

The computing units devoted to collision threat detection and object identification deal with memory transfers and matrix calculations related with neurons models (equations (2), (3), (4) and (5)) and statistical descriptors (equations (9), (10) and (11)), while the general purpose processor deals with low processing demanding scalar calculations of the LGMD model (equations (6), (7) and (8)) and the object classification (equations (12), (13) and (14)).

Both full custom units (figure 3) include a vectorial computing core, with functionality resembling that of standard DSP execution units, containing a vectorial ALU, a multiplier and a barrel shifter, plus two sets of registers for temporal data storage and a programmable sequencer. In addition, a multichannel Direct Memory Access device (DMA), which operates as a stand-alone subsystem, is devoted to image and intermediate data transfers, servicing interrupts when data transactions are complete. The computing cores once programmed, work as peripheral units, processing raw data corresponding to the sensed images, and delivering processed information according to the programmed sequences of instructions.

The computing capabilities of the processing units have been over dimensioned to allow new functionalities and enhancements, as including the speed of the vehicle together with the steering angle within the locust visual neural network model and improving the TpFE object identification capabilities.

Both full-custom units have been carried out in VHDL and are at advanced design stages.

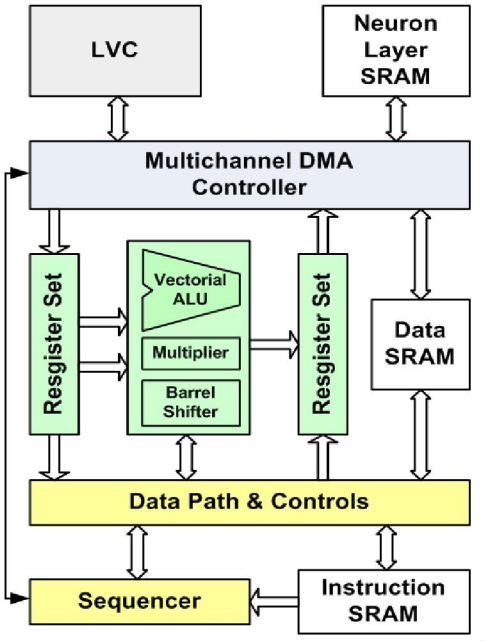

Figure 3. Full custom computing unit block diagram.

V. Locust Vision Chip (LVC)

The Locust Vision Chip (LVC) is composed basically of an array of 150 x 100 retino-topic units. Each unit includes a HDR sensor with a 6-bit in-pixel A/D converter, and the possibility of storing three digital images (as required by the Locust neural network model) at extreme temperatures (110ºC). In addition, the chip provides memory for 4096 words of 32-bit, on-chip generation of all required analog signals and references, digital control, digital communication of acquired images, temperature sensing and test facilities.

The LVC's control commanding requirements include asynchronous and real-time low level control tasks. As mentioned before, a programmable full custom microcontroller has been developed in order to free the external main processor from the low level functions needed to handle the LVC. The architecture and main characteristics of this controller are explained in the next section.

VI. LμC: LVC Microcontroller

The LμC microcontroller is a Reduced Instruction Set Computer (RISC) with fixed encoding register-register Harvard architecture [6] optimized for low area cost and efficient LVC CMOS sensor control. The design of the LμC has been carried out in ASIC-oriented VHDL code [7][8]. In addition, a set of C programs has been developed to automate the generation of VHDL error-free code (corresponding to specific controller modules) and to save design time. Fig. 4 shows a block diagram of the LμC together with the LVC chip.

The microcontroller comprises 16 general purpose registers (GPR), 92 special function registers (SFR), a data path with internal registers, an ALU, a bidirectional shifter, two control units with sequencers one of them accessing a microcode ROM, 4 timers with prescalers for internal/external timing and control purposes and separate data and instruction memory. The device accepts up to 5 interrupt sources.

The LVC low level control and timing tasks are performed via the SFRs and the timers with prescalers. Two sets of 9 byte SFRs include double data and extra addressing ports. These registers have been designed to receive new high level commands from the host CPU, even if the LμC is busy and to send data (coming from the microcontroller's processing routines) to the host CPU.

The GPRs and SFRs share the same memory addressing space to reduce internal data flow. Therefore, ALU operations are directly applicable over the SFRs without using moving-data instructions.

Data memory, GPRs and most SFRs have single data port accessing in order to minimize area cost. Since this fact penalizes instruction speed execution, extra intermediate registers, not accessible by the user, have been included within the data path (Data Path Registers -DPR-) with the aim of decreasing the number of cycles needed for register-register and register-memory data interchange.

To speed-up image uploading and downloading processes, the LμC includes external IO handshaking ports. These ports are handled either with load-store and ALU accessing instructions or with a specific purpose instruction subset which use a minimum-cycle execution stage to save CPU load.

The programming of the LμC is carried out in assembler code. For this objective a two-pass assembler has been coded in C language. The assembler generates output files, coded in hexadecimal, printing format and behavioural VHDL code for post-synthesis simulations which has been used for a full verification of the LμC design.

VII. Conclusion

A VSoC computing architecture suitable for processing information according to the four-layered neural network structure of the locust visual system has been presented. The proposed system includes a HDR CMOS image sensor, low level controlled with a full-custom microcontroller, two full-custom computing units, plus a general purpose processor and is capable of processing images, detecting collision threats and performing elementary object identification in real time automotive scenarios.

In order to test and verify the VSoC, its digital core, now at advanced design stage, will first be synthesized on an FPGA and after being tested and fully verified, included together with the HDR CMOS image sensor within the same ASIC.

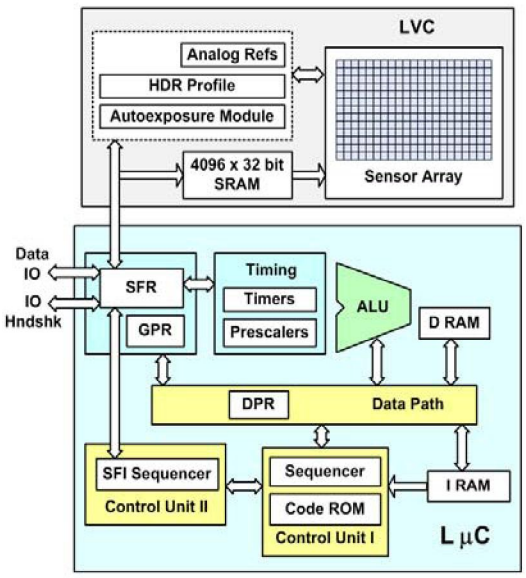

Figure 4. Block diagram of the LμC microcontroller and the LVC sensor.

Acknowledgments

This work has been partially funded by the projects IST2001 – 38097 (LOCUST), TIC2003 – 09817- C02 – 01 (VISTA). Special thanks to Volvo Car Corporation for the test videos used to evaluate the performance of the models.

Ms. Vargas's work is funded by an I3P grant of the Spanish Ministry of Education and Science and the European Social Fund. Mr. Cuadri's work is funded by an F.P.U. grant of the Spanish Ministry of Education and Science.

References

[1] Sonka, Hlavac and Boyle, *Image Processing, Analysis and Machine Vision*. International Thomson Publishing Inc., 1998.

[2] "Intelligent Transportation Systems home page," http://www.its.dot.gov, Avaliable online.

[3] F.C. Rind and P.J. Simmons, "Seeing what is coming: building collision-sensitive neurones," *Trends Neuroscience* 22(5), pages 215-220, 1999.

[4] J.Cuadri, G.Liñán, R.Stafford, M.S.Keil and E.Roca, "A bioinspired collision detection algorithm for VLSI implementation," *SPIE conference on Bioengineered and Bioinspired Systems*, Vol. 5839, pages 238-248, 2005.

[5] S. Yue, F.C. Rind, M.S. Keil, J. Cuadri and R. Stafford, "A bio-inspired visual collision detection mechanism for cars: Optimisation of a model of a locust neuron to a novel environment", *accepted for publication in Neurocomputing*, 2005.

[6] Jonh L. Hennessy, David A. Patterson, *Computer Architecture: A Quantitative Approach*. Morgan Kauffman, 2003.

[7] R. Airiau, J.M. Bergé and V. Olive, *Circuit Synthesis with VHDL*. Kluwer Academic Publishers, 1999.

[8] J. Mermet (Editor), *VHDL for Simulation, Synthesis and Formal Proofs of Hardware*. Kluwer Academic Publishers, 1992.