Using FPGA for visuo-motor control with a silicon retina and a humanoid robot

A. Linares-Barranco, F. Gómez-Rodríguez, A. Jiménez-Fernández.
Arquitectura y Tecnología de Computadores.
Universidad de Sevilla.
Av. Reina Mercedes s/n, 41012-Sevilla, SPAIN
alinares@atc.us.es

T. Delbrück, P. Lichtensteiner
Institute of Neuroinformatics.
UNI-ETH Zürich.
Switzerland

***Abstract*— The address-event representation (AER) is a neuromorphic communication protocol for transferring asynchronous events between VLSI chips. The event information is transferred using a high speed digital parallel bus. This paper present an experiment based on AER for visual sensing, processing and finally actuating a robot. The AER output of a silicon retina is processed by an AER filter implemented into a FPGA to produce a mimicking behaviour in a humanoid robot (The RoboSapiens V2). We have implemented the visual filter into the Spartan II FPGA of the USB-AER platform and the Central Pattern Generator (CPG) into the Spartan 3 FPGA of the AER-Robot platform, both developed by authors.**

I. INTRODUCTION

The Address-Event Representation (AER) was proposed by the Mead lab in 1991 [1] for communicating between neuromorphic chips with spikes (Fig. 1). Each time a cell on a sender device generates a spike, it communicates with the array periphery and a digital word representing a code or address for that pixel is placed on the external inter-chip digital bus (the AER bus). Additional handshaking lines (Acknowledge and Request) are used for completing the asynchronous communication. In the receiver chip the spikes are directed to the pixels whose code or address was on the bus. In this way, cells with the same address in the emitter and receiver chips are virtually connected by streams of spikes. These spikes can be used to communicate analog information using a rate code, but this is not a requirement. Cells that are more active access the bus more frequently than those less active. Arbitration circuits usually ensure that cells do not simultaneously access the bus. Usually these AER circuits are built using self-timed asynchronous logic by e.g. Boahen [2].

Transmitting the cell addresses allows performing extra operations on the events while they travel from one chip to another. For example the output of a silicon retina can be easily translated, scaled, or rotated by simple mapping operations on the emitted addresses. These mapping can either be lookup-based (using, e.g. an EEPROM) or algorithmic. Furthermore, the events transmitted by one chip can be received by many receiver chips in parallel, by properly handling the asynchronous communication protocol. There is a growing community of AER protocol users for bio-inspired applications in vision, audition systems and robot control, as demonstrated by the success in the last years of the AER group at the Neuromorphic Engineering Workshop series [3]. The goal of this community is to build large multi-chip and multi-layer hierarchically structured systems capable of performing massively-parallel data-driven processing in real time [7].

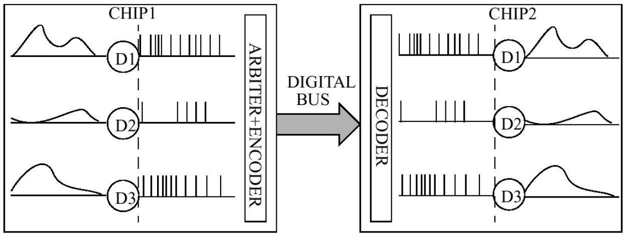

Fig. 1 Rate-coded AER inter-chip communication scheme.

The neuromorphic approach of AER can be also applied to actuators, like the muscles in the biology. In this paper we study a visual processing mechanism to detect the center of an object in movement using an AER retina and we also study the problem of controlling DC motors of a commercial toy, the RoboSapiens V2.

The experiment that we present in this paper is based on connecting the AER output of the TmpDiff128 retina [5] to the USB-AER platform [7] and from there to the AER-Robot platform [6]. The USB-AER FPGA has been programmed with a new visual filter based on the architecture of the frame-grabber tool in [7]. This tool uses a state machine that receives events from the input AER bus and counts the number of events per pixel for a period of time. The tool can for example be used to bin AER output into images or frames for reconstruction. In the present case we added a simple object center detection algorithm to the frame grabber tool that is based on the assumption that leading and trailing edges of a moving object generate opposite types of events. The algorithm calculates the address of the pixel with the maximum activity for both polarities of pixels that the retina returns (on-cells and off-cells) and returns the center point between these two pixels.

The AER-Robot interface receives through the AER bus the computed position of the center of the object. According to the coordinates, the AER-Robot FPGA will signal to the DC motors to move the arms and the hip of the robot. This way the RoboSapiens V2 will mimic the movement of the object that is being seen by the retina. Fig. 2 shows a block diagram of the experiment.

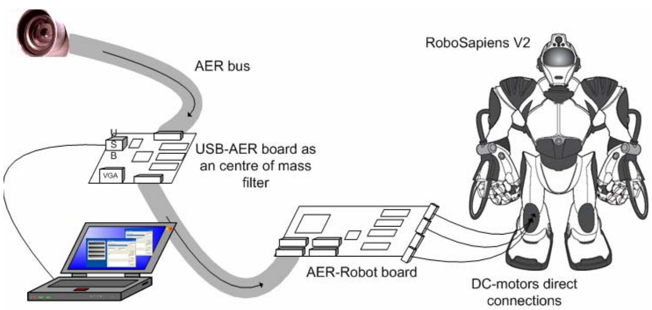

Fig. 2 Block diagram of the Visuo-motor experiment.

In the following sections we discuss the tmpdiff128, the visual filter and the CPG.

II. SILICON RETINA.

The main properties of the transient vision sensor are summarized in Table 1 and Fig. 3. Each address-event signifies a change in logarithmic intensity as given in (1)

$$\left|\Delta \log I\right| > T \tag{1}$$

where I is the pixel illumination and T is a global threshold. Each event thus means that log$I$ changed by T since the last event and specifies in addition the sign of the change. Thus events generally encode scene reflectance changes. Because this computation is based on a very compressive logarithmic transformation in each pixel, it also allows for wide dynamic range operation (120 dB, compared with e.g. 60 dB for a high quality traditional image sensor). This wide dynamic range means that the sensor can be used with uncontrolled natural lighting. The asynchronous response property also means that the events have the timing precision of the pixel response rather than being quantized to the traditional frame rate. Thus the “effective frame rate” is typically several kHz. If the scene is not very “busy”, then the data rate can easily be a factor of 100 lower than from a frame-based image sensor of equivalent time resolution. The unique design of the pixel also allows for unprecedented uniformity of response. The mismatch between pixel contrast thresholds is a modest 2.1%, so that the pixel event threshold can be set to a few percent contrast, allowing the device to sense real-world contrast signals rather than only artificial high contrast stimuli. The vision sensor also has integrated digitally-controlled biases that greatly reduce chip-to-chip variation in parameters and temperature sensitivity. And finally, the system we built has a standard USB2.0 interface that delivers time-stamped address-events to a host PC. This combination of features has meant that we have had the possibility of developing algorithms for using the sensor output and testing them easily in a wide range of real-world scenarios.

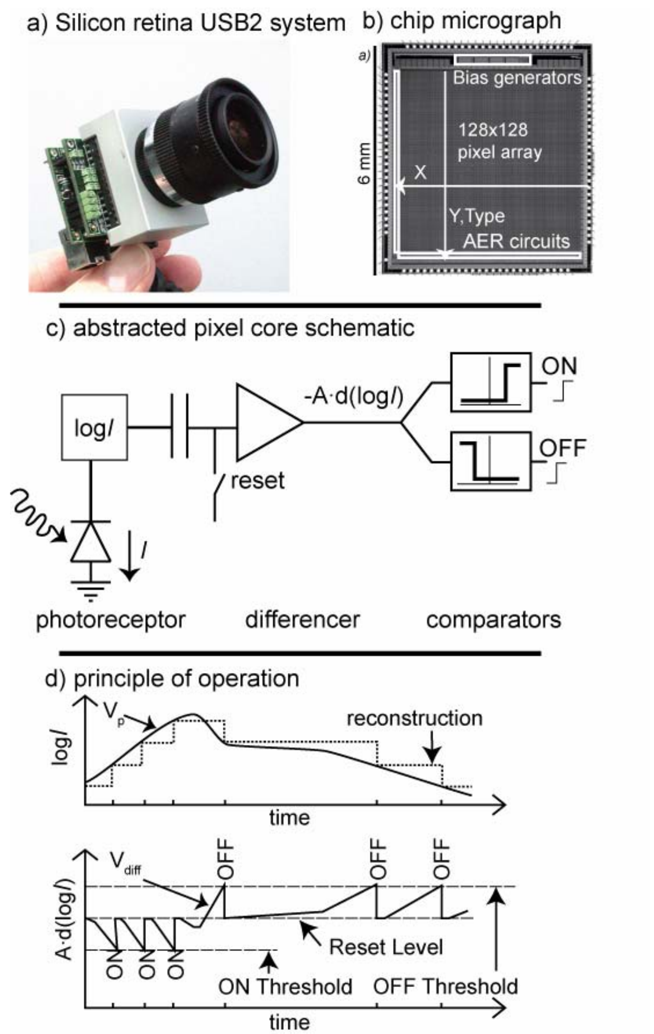

Fig. 3 Summarizes characteristics of Tmpdiff128 transient vision sensor. a) shows the vision sensor with its lens and USB2.0 interface; b) shows a die photograph labeled with the row and column from a pixel that generates an event with x,y,type output; c) shows an abstracted schematic of the pixel that responds with events to fixed-size changes of log intensity; d) illustrates how the ON and OFF events are internally represented and output in response to an input signal. Figure adapted from [1].

Table 1 Tmpdiff128 transient vision sensor specifications

Functionality	Asynchronous temporal contrast
Pixel size um (lambda) *Fill factor (%)*	40x40 (200x200) 9.4%%
Fabrication process	4M 2P 0.35um
Pixel complexity	26 transistors (14 analog), 3 capacitors
Array size	128x128

Die size mm²	6x6.3
Interface	15-bit word-parallel AERs
Power consumption	24mW @ 3.3V
Dynamic range	120dB 2 lux to > 100 klux scene illumination with f/1.2 lens
Response latency *Events/sec*	15µs @ 700mW/m² ~1M events/sec
FPN, matching	2.1% contrast

III. AER VISUAL FILTER FOR FPGA.

This section describes how we filter the retina output to obtain the center of mass of the moving object.

The USB-AER board collects events, calculates the pixel with maximum 'on' traffic and the one with maximum 'off' traffic. The center of mass of the object seen by the retina is computed simply by calculating the center point between both positive and negative maximum traffic points. To avoid errors produced by the retina when the traffic is low (because of ambient changes, like fluorescent lights) a threshold is included for calculating 'on' and 'off' minimum traffic. This calculation is made every 1.5 ms.

Fig. 4 shows data from the retina in response to a moving ball (top row) and the output of the filter (bottom row) in response to the retina data. The images show slices of 20ms integrated output. Due to this relatively long time window, several events are visible even though the filter output is a single pixel at all times.

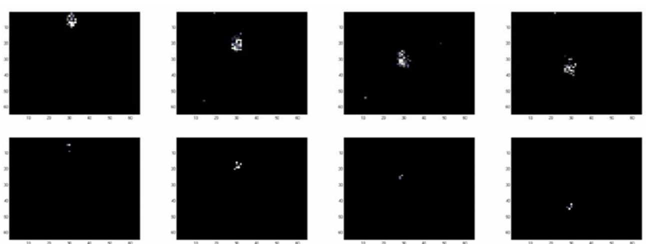

Fig. 4 Filter images. The top row shows the input of the filter, the bottom row show the output of the filter. For visualization the bottom row is obtained by intergrating events during 20 ms.

IV. AER CPG

This section describes the implementation of the Central Pattern Generation based on AER. The CPG receives as input the coordinate of the computed center of the object. The CPG produces an upward movement of the left robot arm in response to an upward movement of the center of an object detected in the left half of the visual field of the retina. Analogously the CPG produces downward movements of the left arm and up and down movements for the right arm. The CPG is implemented in VHDL and tested on an AER platform called AER-Robot (shown in Fig. 5 and Fig. 6). This new platform is an improved version of a prototype designed to control an anthropomorphic AER hand [6]. The platform is designed around a Spartan 3 400 FPGA with 4 parallel AER connectors (2 input and 2 output), 4 power stages to manage 4 DC motors with two encoder channels and 4 hall effect current sensor to measure the power consumption of the motors. The interface also has 12 analog sensor inputs and 36 general purpose digital ports.

With this FPGA, the interface is able to receive high AER rates, process them together with the input from the robot sensors and encoders and control the motors of the robot.

The platform was developed as an interface between AER systems and robots using two AER buses: one for incoming events and another for outgoing information (events) about the state of the motors and the sensors. The input AER bus can be replicated into the output AER bus, called AER IN pt, to conveniently allow a chain of several boards connected by the AER buses.

The CPG implemented in this platform make a decision about which arm to move and in which direction based on the trajectory of the last N events received from the AER visual filter implemented into the USB-AER platform. It could be possible that the noise produced by the retina due to visual noise around the object (like a fluorescent tube), could imply virtual objects detection by the visual filter that doesn't follow a real trajectory. This behavior is filtered by the CPG by monitoring a number of events and ordering the movement of the arm if the trajectory is feasible.

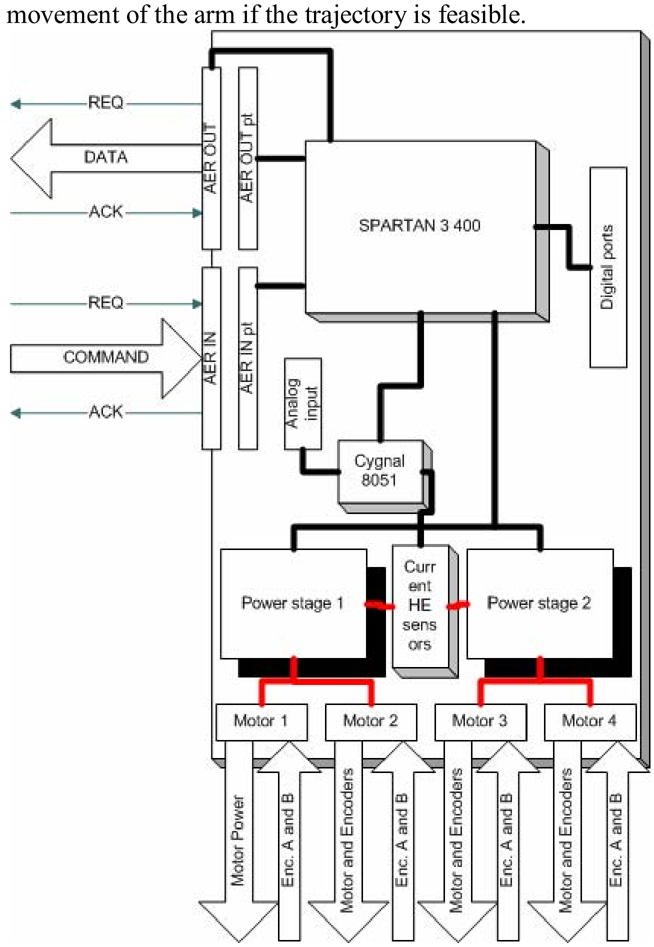

Fig. 5 Block diagram of the AER-Robot platform.

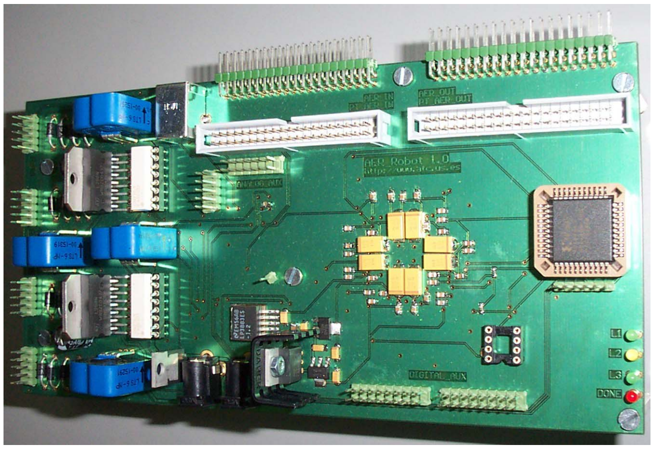

Fig. 6 AER-Robot board photograph.

Fig. 6 shows a photograph of the AER-Robot Interface PCB. The digital part of the PCB is in the middle. The board has a Cygnal 80C51F320 microcontroller for the analog to digital conversion (200Ksamples/second and 10-bits) of the sensor measurements and a USB port for the PC connectivity.

The VHDL of the CPG is divided in two parts: (a) PWM (pulse width modulation) generation and (b) movement decision.

(a) PWM. This part of the VHDL generates a PWM signal for the DC motors in the direction and intensity configured by part (b). The PWM signal modulates the analog intensity of the DC motor using a digital periodic signal with two parameters: the period and the time that the signal is high for each period of time. A counter manages the period time of the PWM signal and a comparator controls the high time of the signal.

(b) This part receives a sequence of events from the filter. These events are coordinates of the center of the object detected. To be sure that the center comes from an object and it is not the result of noise in the AER retina, the filter only gives coordinates to the output if the traffic is higher than a threshold. The received coordinates are preprocessed to calculate the direction (up or down) of the movement. For the preprocessing, the last 4 events received are used. If the last 4 events indicate identical direction of movement, an order is given to part (a) to initiate a movement, else, a stop order is given to part (a). The possible trajectories are a left movement, a right movement and an up or down movement separated for the two halves of the visual field. The left and right movements are sent to the hip DC motor, and the up and down movements to the left and right arms. Fig. 7 shows this decision process.

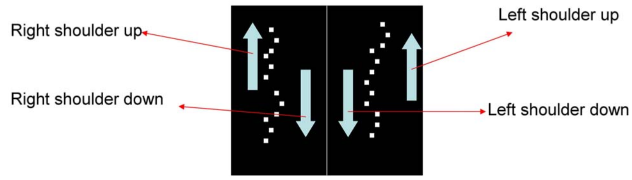

Fig. 7 CPG decision process for y-axis. If the moving pixel is in the right part of the scene, a signal to move the left arm will be sent. If the moving pixel is going up, a signal to move up the arm will be sent.

V. CONCLUSIONS

We showed how a retina with a FPGA AER based processing can efficiently be used to process AER data in real time for object detection and robot movement. The whole system comprising an AER retina, two custom-made AER processing platforms and a humanoid toy robot was successfully tested during the Telluride Neuromorphic Engineering Workshop 2006. The AER-Robot platform has capacity enough to implement both the vision filter and the CPG in the same FPGA.

ACKNOWLEDGMENTS

This work was supported by EU grant IST-2001-34124 (CAVIAR) and the Spanish grant TIC-2003-08164-C03-02 (SAMANTA). We would also like to thank the NSF sponsored Telluride Neuromorphic Engineering Workshop for bringing the authors together to work on this project.

REFERENCES

[1] M. Sivilotti, Wiring Considerations in analog VLSI Systems with Application to Field-Programmable Networks, Ph.D. Thesis, California Institute of Technology, Pasadena CA, 1991.

[2] Kwabena A. Boahen. “Communicating Neuronal Ensembles between Neuromorphic Chips”. Neuromorphic Systems. Kluwer Academic Publishers, Boston 1998.

[3] A. Cohen et al., *Report to the National Science Foundation: Workshop on Neuromorphic Engineering*, Telluride, Colorado, USA, June-July 2004. [www.ini.unizh.ch/telluride]

[4] Misha Mahowald. VLSI Analogs of Neuronal Visual Processing: A Synthesis of Form and Function. PhD. Thesis, California Institute of Technology Pasadena, California, 1992.

[5] P. Lichtsteiner, et al., "A 128×128 120dB 30mW Asynchronous Vision Sensor that Responds to Relative Intensity Change," ISSCC Dig. of Tech. Papers, San Francisco, 2006, pp. 508-509 (27.9).

[6] A. Linares-Barranco et al.. “AER Neuro-Inspired interface to Anthropomorphic Robotic Hand”. IEEE World Conference on Computational Intelligence. International Join Conference on Cellular Neural Networks. Vancouver, July-2006.

[7] R. Serrano-Gotarredona et al. AER Building Blocks for Multi-Layer Multi-Chip Neuromorphic Vision Systems. NIPS 2005. Vancouver.